Relational Outlier Detection: Techniques and Applications

Yen-Cheng Lu

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Computer Science

Chang-Tien Lu, Chair
Ing-Ray Chen
Chandan Reddy
Alireza Haghighat
Feng Chen

May 5th, 2021
Falls Church, Virginia

Keywords: Relational Outlier Detection, Generalized Linear Model, Robust Estimation, Music Genre Recognition, Time Series Outlier Detection

Relational Outlier Detection: Techniques and Applications

Yen-Cheng Lu

ABSTRACT

Nowadays, outlier detection has attracted growing interest. Unlike typical outlier detection problems, relational outlier detection focuses on detecting abnormal patterns in datasets that contain relational implications within each data point. Furthermore, different from the traditional outlier detection that focuses on only numerical data, modern outlier detection models must be able to handle data in various types and structures. Detecting relational outliers should consider (1) Dependencies among different data types, (2) Data types that are not continuous or do not have ordinal characteristics, such as binary, categorical or multi-label, and (3) Special structures in the data. This thesis focuses on the development of relational outlier detection methods and real-world applications in datasets that contain non-numerical, mixed-type, and special structure data in three tasks, namely (1) outlier detection in mixed-type data, (2) categorical outlier detection in music genre data, and (3) outlier detection in categorized time series data.

For the first task, existing solutions for mixed-type data mostly focus on computational efficiency, and their strategies are mostly heuristic driven, lacking a statistical foundation. The proposed contributions of our work include: (1) Constructing a novel unsupervised framework based on a robust generalized linear model (GLM), (2) Developing a model that is capable of capturing large variances of outliers and dependencies among mixed-type observations, and designing an approach for approximating the analytically intractable Bayesian inference, and (3) Conducting extensive experiments to validate effectiveness and efficiency.

For the second task, we extended and applied the modeling strategy to a real-world problem. The existing solutions to the specific task are mostly supervised, and the traditional outlier detection methods only focus on detecting outliers by the data distributions, ignoring the input-output relation between the genres and the extracted features. The proposed contributions of our work for this task include: (1) Proposing an unsupervised outlier detection framework for music genre data, (2) Extending the GLM based model in the first task to handle categorical responses and developing an approach to approximate the analytically intractable Bayesian inference, and (3) Conducting experiments to demonstrate that the proposed method outperforms the benchmark methods.

For the third task, we focused on improving the outlier detection performance in the second task by proposing a novel framework and expanded the research scope to general categorized time-series data. Existing studies have suggested a large number of methods for automatic time series classification. However, there is a lack of research focusing on detecting outliers from manually categorized time series. The proposed contributions of our work for this task include: (1) Proposing a novel semi-supervised robust outlier detection framework for categorized time-series datasets, (2) Further extending the new framework to an active learning system that takes user insights into account, and (3) Conducting a comprehensive set of experiments to demonstrate the performance of the proposed method in real-world applications.

Relational Outlier Detection: Techniques and Applications

Yen-Cheng Lu

GENERAL AUDIENCE ABSTRACT

In recent years, outlier detection has been one of the most important topics in the data mining and machine learning research domain. Unlike typical outlier detection problems, relational outlier detection focuses on detecting abnormal patterns in datasets that contain relational implications within each data point. Detecting relational outliers should consider (1) Dependencies among different data types, (2) Data types that are not continuous or do not have ordinal characteristics, such as binary, categorical or multi-label, and (3) Special structures in the data. This thesis focuses on the development of relational outlier detection methods and real-world applications in datasets that contain non-numerical, mixed-type, and special structure data in three tasks, namely (1) outlier detection in mixed-type data, (2) categorical outlier detection in music genre data, and (3) outlier detection in categorized time series data. The first task aims on constructing a novel unsupervised framework, developing a model that is capable of capturing the normal pattern and the effects, and designing an approach for model fitting. In the second task, we further extended and applied the modeling strategy to a real-world problem in the music technology domain. For the third task, we expanded the research scope from the previous task to general categorized time-series data, and focused on improving the outlier detection performance by proposing a novel semi-supervised framework.

Acknowledgments

My deep gratitude to my advisor Dr. Chang-Tien Lu for his research guidance, for believing in me, motivating me to stay in the program as long as it took, and encouraging me to complete my study. I am thankful to my committee members, Dr. Chandan Reddy and Dr. Alireza Haghighat, for their insightful help, Dr. Ing-Ray Chen for his keen personal interest and encouragement, and Dr. Feng Chen for all the research guidance and close collaborations. I am grateful to my supervisor Dr. Jason Tao and project manager Rakesh Nune at the DC Department of Transportation for their support and encouragement. I greatly appreciate the support and help of my lab colleagues Dr. Kaiqun Fu, Dr. Zhiqian Chen, Dr. Xutong Liu, Dr. Manu Shukla, and all the close friends at the lab. I also want to thank my long-time friend Dr. Chih-Wei Wu for supporting and sharing research insights with me. Finally, I dedicate this dissertation to my classmate, best friend and wife, Dr. Yating Wang, who sacrificed numerous weekends and evenings and kept me pursuing the completeness of this work.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In recent years, with the flourishing of smart devices and service-oriented applications in the industry, pattern recognition in various types of datasets became an important research topic. Outlier detection, one of the most crucial research domains in machine learning and data mining communities that focuses on identifying the anomalous points in the datasets, became more and more crucial. Outlier detection techniques have been widely applied in a variety of domains, such as health monitoring [1], cyber security [2], military surveillance [3], and financial systems [4]. Although it has been studied for decades, automatically detecting outliers is still an unsolved issue when the data volume and variety become larger.

Outlier detection is important because outliers are often related to abnormal instances in the datasets, which are usually the most critical points that demand caution or alarm. For example, in the cybersecurity domain, an outlier can be an intruder that performs malicious actions that are different from the behavior of normal users. In credit card transaction datasets, an outlier can be identified by a transaction at a very different location or an abnormal volume of transactions that imply potential fraud events. In a transportation monitoring scenario, a traffic volume spike in the off-peak hours is often related to the occurrence of traffic accidents or special events that require special traffic controls. Similarly, in public health data, an abnormal combination of symptoms and body conditions that affect an unusual number of patients may refer to a new disease outbreak. Sometimes, outliers may have positive meanings. For example, in the art, literature, and music technology domain, outlier detection techniques are applied to detect novel creations; the specific application is also referred to as novelty detection.

Most existing outlier detection approaches focus on detecting abnormal data instances without considering the inner relationship among the features. However, in many real-world datasets, the relationships among the features of each data instance are more informative than the features themselves. Figure 1.1 describes an example in the housing market dataset. For a typical outlier detection approach, Instance 4 will most likely be identified as the outlier, given it has an extreme value in the feature space. However, Instance 5 presents a more interesting abnormal behavior because its attributes-to-price relation is very different from the other instances in the same dataset. This type of outlier is usually hidden and difficult to identify. This research focuses on the development of relational outlier detection methods that can detect relational outliers in non-numerical type and special structured data. The work demonstrates the effectiveness of the approaches in real-world applications. More specifically, this work proposes a solution for research issues of outlier detection in mixed-type datasets, categorical outlier detection in music genre datasets, and detecting outliers in categorized time-series data. These tasks have a wide array of applications. Some of them are described below.

Id	sq ft.	# Rooms	# Floors	Age	HWY in 3 miles	List Price
1	2300	5	3	20	No	$ 600,000
2	1100	2	2	10	No	$ 300,000
3	550	1	1	10	Yes	$ 250,000
4	15580	20	3	20	Yes	$ 5,000,000
5	2100	4	2	10	Yes	$ 280,000
6	2150	5	3	30	No	$ 500,000
7	1260	2	2	15	Yes	$ 550,000

Figure 1.1: Relational Outlier in a Housing Market Dataset

- **Virus detection in a computer network.** With the increasing use of internet applications, many computers are exposed to the risks of a virus or intruders spreading through the network [5]. The virus usually performs a series of actions to gain control of a computer or damage a system.

- **Audio and video genre classification.** The literature on genre classification in audio and video files has been growing for several decades. Fisher et al. proposed the earliest movie

genre classification system [6], and Tzantakis et al. proposed a classic automatic music genre classification system [7]. Genre can be recognized by various approaches. For example, Ivasic-kos proposed an approach to identify movie genres from posters [8]. In recent years, the prevalence of online streaming services has attracted more attention to this issue because a sophisticated genre classification system could be adapted and applied for music and movie recommendations and their business purposes.

- **Inappropriate content detection.** Inappropriate content prohibition is an important issue due to the prevalence of the internet. Many inappropriate content detection system are proposed, mainly focused on pornographic video detection [9, 10, 11, 12]. However, the detection of violent and bloody contents is comparatively less studied [13]. This application is an extension of genre classification and outlier detection. When the feature set captures the concept pattern of normal video clips in training data, the clips that have inappropriate content will be identified as outliers.

- **Fraud and financial crime detection.** Data mining techniques are also widely applied to financial and crime analysis [14, 15, 16]. Like malicious behaviors in computer networks, most financial fraud, such as money laundering and credit card fraud, is a series of abnormal actions. Time-series outlier detection methods can be applied for this objective.

- **New disease detection and pattern recognition.** Detecting new diseases in the early stage of an outbreak has been studied for decades in medical and bioinformatics domains. This application does not demand instant data processing but does require a result with high accuracy and the capability of mining patterns in big data. While most medical clinical reports are mixed-type data, a well-designed, mixed-type outlier detection method can improve the accuracy of identifying suspicious instances and help medical professionals be aware of their patients' hidden abnormalities.

1.1 Research Issues

This research aims to investigate and develop outlier detection methods for identifying relational outliers in datasets. The major research issues are stated as follows:

1.1.1 Outlier Detection in Mixed-type Datasets

The majority of existing outlier detection techniques are developed for single-type datasets, while the majority of real-world datasets contain a mixture of data types that may be continuous or isolated, structured or unordered. Applying single-type outlier identification techniques to mixed-type data reduces precision because no unified measure exists to capture the dependencies between various types of attributes. For instance, while a distance-based approach for numerical data involves the distance between data instances to determine anomalies, no such distance of the same measure exists for a categorical data attribute. A few methods are planned specifically for mixed-type files, including LOADED [17] and RELOADED [18]. Both focus on increasing numerical efficiency, and their methods for capturing associations between mixed-type attributes are heuristic in nature and lack a firm statistical foundation. Three main challenges for mixed-type anomaly detection are listed below:

1. **Modeling correlations among mixed-type attributes:** Because mixed-type datasets have mixed dimensions of various types of attribute dependence, the relationships between these attributes in multivariate and different types must be modeled.;

2. **Capturing large variations introduced by anomalies:** To learn what constitutes natural behavior, the majority of existing approaches require a pure testing dataset. However, identifying normal instances is difficult for unsupervised frameworks in the midst of deviations, since these outliers produce large deviations that can potentially skew the estimate of normal behavior patterns;

3. **Analytically intractable posterior inference:** The joint likelihood of the non-Gaussian objectives creates an analytically intractable distribution.

In this work, we present an outlier detection method in this work using a novel regression-based model that is combined with a generalized linear model (GLM) and rigorous estimation techniques. Two Bayesian inference methods are presented for the purpose of resolving intractable inferences. An extensive set of experiments was performed on both synthetic and real-world data to determine the proposed model's efficacy and performance.

1.1.2 Categorical Outlier Detection in Music Genre Datasets

Automatic music genre recognition (MGR) is a hot research subject in the field of music information retrieval (MIR). Hundreds of methods have been proposed over the decades to develop music genre identification approaches. Tzantakis et al. [7] proposed a classic system, along with the most common MGR dataset in subsequent research works, namely the *GTZAN* dataset. Although *GTZAN* became the de facto normative baseline dataset for a decade, Sturm states in [19] that the dataset is far from perfect. According to Sturm, almost fifty music clips spanning ten genres are apparently distorted or incorrectly labeled, and even more clips are debatably labeled with their respective genre classes.

Outlier detection has been researched and implemented in a wide range of application domains for decades in the machine learning community; nevertheless, the method has been applied sparingly in the area of MIR. We present a novel model for automatically identifying incorrect data in MGR datasets in this work. We present and analyze some existing anomaly detection techniques on the music datasets. Besides, we also studied the capability of how well the existing algorithms can detect misclassified clips and corrupted files, and the potential that datasets can be improved in terms of accuracy. The empirical findings indicate that our proposed approach outperforms other known methods for this purpose, but it still needs refinement.

1.1.3 Detecting Outliers in Categorized Time-Series Data

Continued from the second task, this research task extended the scope and focuses on developing an outlier detection method for a special structure dataset that is often introduced by time-series classification and clustering problems. Examples include video classification by various criteria such as genre, action, language, and others. The objective of this task is to detect the time series that have been assigned to the wrong category. The challenges are:

1. **Modeling the temporal dependencies in time series:** This task requires a modeling approach that captures the temporal dependencies of the time series and provides a similarity measure between each pair of the time series. The existing approaches that capture temporal dependencies, including auto-regressive models [20], HMM-based models [21], and RNN-based [22, 23] models, are mostly not feasible for this task due to their complexity and lack

of similarity measure to time-series pairs.

2. **Capturing contributor relations and cross-category relationships:** The categories of each time series are contributed by the features. Traditional outlier detection methods discover the outliers by identifying the abnormal pattern of the features within each category, while cross-category relationships are typically ignored. However, feature-category and cross-category relationships provide important information regarding how features contribute to the decision of category and the similarity or confusion level among the categories.

3. **Robust modeling:** A model without a robust design can be easily biased by a small number of outliers. Specifically, time-series data usually have large variance due to their larger size and variety. Designing a robust model that captures the pattern from data with large variance and resists outliers from distorting the captured pattern thus becomes a challenging issue.

1.2 Contributions

The major proposed research contributions can be stated as follows:

Generalized Outlier Detection in Mixed-type Data

- **Establishing a novel unsupervised framework**: We propose a novel unsupervised model that is able to perform general-purpose relational outlier detection on datasets that contain mixed-type attributes. The proposed framework does not require any outlier labels, which is usually difficult to obtain in reality.

- **Capturing dependencies among mixed-type observations and large variances of anomalies**: The new model solves both primary difficulties associated with identifying anomalous instances in a mixed-type model, namely modeling mutual correlations in the feature spaces of mixed-type data and capturing the significant variations caused by anomalies.

- **Designing effective solutions for approximating Bayesian inference**: Two approaches to Bayesian inference are proposed: one is based on integrated nested Laplace approximation (INLA), and the other is adapted from expectation propagation (EP) using variational-EM method.

- **Conducting comprehensive experiments to demonstrate the efficacy and efficiency**: The experimental findings show that the proposed framework performed better than the majority of benchmark methods on both synthetic and real-world datasets while maintaining reasonable computational performance. Additionally, an experimental study is used to determine the advantages and disadvantages of the proposed approaches.

Categorical Outlier Detection in Music Genre Datasets

- **An unsupervised anomaly detection approach in music genre datasets:** We proposed an unsupervised method for discovering anomalies in music genre datasets. There is no need for an outlier label.
- **A categorical outlier detection model:** The robust regression-based modeling approach is generalized to support the identification of categorical anomalies. Additionally, a novel method for approximating analytically intractable inference is derived.
- **An experimental study applying state-of-the-art outlier detection methods to MGR datasets:** A comprehensive experimental analysis is performed on the most commonly used MGR databases. Five developed methods for detecting outliers are introduced and extended to the datasets. The proposed solution outperforms current state-of-the-art approaches, as shown by the results.

Detecting Outliers in Categorized Time-Series Data

- **Constructing a novel semi-supervised framework**: A novel semi-supervised framework capable of performing general-purpose outlier detection on time-series classification (TSC) data is proposed. In contrast to the traditional outlier detection algorithms that often only focus on the anomalous measures, the proposed framework aims to enhance the capability of recognizing the normal behavior by expanding the pattern defined by a set of seed normal instances.
- **Designing a robust outlier detection mechanism**: The proposed framework can tolerate outliers and remain unbiased when noise is present in the datasets. With the multi-arbitrator mechanism, one biased model will not significantly impact the overall expansion process.

- **Adapting an active learning strategy**: As an alternative, the semi-supervised framework has also been extended to an active learning-based framework for interactively considering user feedback in the model fitting process.
- **Validated the effectiveness and efficiency by extensive experiments**: Our comprehensive experiments demonstrate the efficacy of the new approach on five real-world datasets from different domains. The advantages and limitations of the proposed approach are also analyzed through a set of experiments.

1.3 Thesis Organization

The remainder of this research proposal is organized as follows. Chapter 2 defines the generalized outlier detection framework for mixed-type data, designs its corresponding model, derives the inference approximation methods, and presents experimental results on synthetic and real-world data. Chapter 3 describes the proposed outlier detection application in MGR datasets, discusses the model and inference methods, and demonstrates the empirical results. Chapter 5 illustrates the research plan for this proposal together with the schedule and current publications.

Chapter 2

Outlier detection in mixed-type data

Detecting outliers in mixed-type data is an important issue that has not been well addressed in the data mining research domains. Many current approaches are based on heuristics and lack a statistical basis, and their modeling of correlation between mixed-type attributes is heuristic. MIxed-Type Robust dEtection (MITRE) is a robust error buffering technique for anomaly detection in mixed-type datasets that we propose in this work. Due to the non-Gaussian design, the exact inference becomes analytically intractable. To solve this problem, two Bayesian inference methods are adopted and derived: integrated-nested Laplace approximation (INLA) and expectation propagation (EP) with variational expectation-maximization (VEM). A series of optimizations is developed to improve computational efficiency. Furthermore, MITRE's effectiveness and efficiency were tested through a large set of experiments using both synthetic and real-world data. This work has been published in earlier articles [24][25].

2.1 Introduction

Outlier detection is a critical issue which has received a lot of attention in decades. The aim is to automatically identify the anomalous behavior and recognize odd occurrences, which are referred to as outliers. For example, in signal processing, outliers could be triggered by spontaneous hardware issues or sensor failures, while outliers in a bank transaction dataset may indicate a fraudulent activity. Outlier detection approaches have been commonly used in a number of fields, including

cybersecurity [2], health monitoring [1], financial systems [4], and military surveillance [3].

Anomaly detection approaches can be categorized into distance-based [26][27], one-class classifier-based[28][29], density-based [30][31], and statistical model-based methods [32][33][34]. The majority of these methods are optimized for datasets that consist of numerical vectors, whilst the majority of real-world datasets contain a mix of data types, including numerical, binary, ordinal, nominal, count, and others. In the KDD panel discussion [35] and the resulting position paper [36], handling mixed-type data was nominated as one of the top ten critical challenges for the next decade in data mining research. However, as single-type methods are directly applied to mixed-type data, major associations between attributes are lost, and their expansion to mixed-type data is technically difficult. For instance, distance-based approaches depend on a well-defined measure to determine the proximity of data instances but lack a global measure for mixed-type attributes, whereas statistical-based approaches model the correlations among the numerical attributes but lack a global correlation measure for mixed-type attributes.

The few approaches available for handling mixed-type data, such as LOADED [17] and RELOADED [18], all prioritize computational performance over the performance metrics. They also use heuristic approaches to capture the correlation among mixed-type attributes, which is lacking a solid theoretical foundation.

Three major challenges for detecting outliers in mixed-type data are: **(1) Modeling correlations among mixed-type data attributes:** Because mixed-type datasets have mixed dimensions of various types of attribute dependence, the relationships among these data attributes in multivariate and different types must be modeled.; **(2) Capturing large variations introduced by anomalies:** To learn what constitutes natural behavior, the majority of existing approaches require a pure testing dataset. However, identifying normal instances is difficult for unsupervised frameworks in the midst of deviations, since these outliers produce large deviations that can potentially skew the estimate of normal behavior patterns; and **3) Inference of the analytically intractable posterior distribution:** The joint likelihood of the non-Gaussian objectives creates a complicated distribution, which is analytically intractable. In order to estimate the posterior distribution, an Bayesian approximation approach is required to approximate the inference for a particular set of observations.

The present work proposes a statistical method for addressing these challenges. We begin by introducing a novel variation of the generalized linear model (GLM) that is capable of capturing the mutual correlations among the attributes in mixed-type. Specifically, the values from the mixed-type

attributes are hashed to a new space consists of hidden numerical random variables that are naturally distributed with a multivariate Gaussian distribution. Each attribute is associated with a hidden numerical variable through a defined link function, for example, a logit function for binary attributes or a log function for count attributes.

With this strategy, the dependency among the mixed-type attributes can be captured using a variance-covariance matrix that tracks the relationship between their latent variables. Meanwhile, an "error buffer" variable assumed with a heavy-tailed Student-t distribution is utilized to account the large variations introduced by outliers. The "error buffer" is designed to absorb all errors while fitting the data into the model. After model fitting, the outlier detection process accesses this error buffer and identifies those instances with unusual magnitudes of error.

As a result of the combined application of GLM and the Student-t assumption on the prior, the exact inference is no longer analytically tractable. Therefore, we present a method that applies the INLA algorithm as the optimization strategy to approximate the Bayesian inference. As an alternative solution, we also proposed a framework that incorporates EP [37] and VEM.

The main contributions of this work are as follows:

1. **Establishing a novel unsupervised framework**: We propose a novel unsupervised model that is able to perform general-purpose relational outlier detection on mixed-type datasets. The new framework does not need any labeled data, which is usually difficult to obtain in reality.

2. **Capturing the dependencies among mixed-type observations and large variances caused by anomalies**: The proposed method solves both of the primary challenges of detecting anomalies in a mixed-type model, namely, modeling reciprocal correlations among the mixed-type attributes and capturing significant variations that are introduced by anomalies.

3. **Designing more effective approaches for Bayesian inference approximation**: Two approaches to Bayesian inference are proposed: one is based on INLA, and the other is adapted from EP using a VEM framework.

4. **Conducting comprehensive experiments to validate the effectiveness and efficiency**: Our experimental findings show that the proposed methods outperformed the benchmark approaches on both synthetic and real-world datasets while maintaining reasonable computa-

tional performance. Additionally, an experimental study is used to determine the advantages and disadvantages of the proposed approaches.

The remainder of this work is divided into the following parts. Section 2.2 summarizes the current literature in this area, while Section 2.3 discusses the problem formulation and model construction. Section 2.4 discusses the architecture for the anomaly detection process, while Section 2.5 presents tests on both virtual and actual datasets. Section 2.6 ends with a rundown of the results and our conclusions.

2.2 Related Work

This section summarizes recent literature on outlier detection, including techniques for detecting single-type and mixed-type anomalies.

Single-type Outlier Detection: The early research on outlier detection are classified into five different categories: distance-based [26][27], density-based [30][31], cluster-based [38], classification-based [28][29], and statistical-based [32][33][34][39] methods.

Knorr et al. [26] introduced the first distance-based method for detecting deviations by the application of a distance threshold. Ramaswamy et al. [27] suggested another early distance-based approach by integrating the distance criteria with the k-nearest neighbor (KNN) method. Although these types of methods are often effective, their precision suffers when the data distribution is distorted. Apart from the distance-based methods, various density-based methods were also proposed. For instance, local outlier factor (LOF) [30] proposed by Breunig el al. and local correlation integral (LOCI) [31] proposed by Papadimitriou et al. These methods focus on inferring the outlierness based on local densities around the targeting instances and their neighbors.

Other approaches to outlier detection frame the issue as a conventional data mining problem. The clustering-based approach suggested in [38] groups related data first and then marks instances that do not cluster well as anomalies. Numerous classification-based methods have also been suggested, assuming that a classification algorithm can learn how to classify anomalies.

This approach is shown by Das et al. [28], who propose a method based on the one-class SVM classifier, and Roth [29], who propose a kernel Fisher discriminant-based approach.

Statistical methods presuppose a particular distribution for the data and classify exceptions by finding instances of low likelihood densities. The well-known masking and swamping effects are one of the primary challenges here. Anomalies may introduce bias into the calculation of distribution parameters, resulting in erroneous likelihood densities that misidentify regular artifacts as anomalies or vice versa.

A variety of methods that make different distribution assumptions have been suggested to address this problem, including techniques based on direction density ratio estimation [33], the robust Mahalanobis distance [32], and the minimum covariance determinant estimator [34]. Recent developments have largely focused on the application of robust statistics to anomaly detection [39].

Another technique often used to find outliers is robust principle component analysis (PCA) [40][41][42][43]. These approaches are well-suited for extracting the most important features from noisy datasets because they are either powered by accurate statistics, such as excluding extreme observations [40] using median values rather than mean values [41], or they decompose the data instances into a low-rank matrix or a sparse matrix [42][43]. In the first example, anomalies are the data instances that have attributes deviate from a predefined threshold, while in the second case, anomalies are the data instances whose sparse matrix has any greater value.

Mixed-type Outlier Detection:

In the real world, data are typically a combination of data types, with non-numerical data exhibiting characteristics distinct from numerical data. For example, because categorical data are not ordered, it is impossible to calculate the variations between data points [44]. This suggests that detection methods designed for numerical data will not always be a good match for mixed-type datasets. Tran et al. [45] model heterogeneous datasets using restricted Boltzmann machines, in which latent binary variables catch the dependency between data fields. Although their solution can be used as a classifier for discrete outputs or as a regression method for continuous outputs, it does not directly account for any dataset anomalies.

According to recent literature, a popular method is to process different data types independently and then merge the outcomes for each data type to detect abnormalities [17][18][46][47][48]. Otey et al. introduced two methods for detecting mixed-type anomalies: LOADED [17] and RELOADED [18]. LOADED calculates the support count of item sets for categorical attributes using an augmented lattice and then computes a correlation matrix for numerical attributes using a correlation matrix. It identifies anomalies by awarding an anomaly score dependent on the item sets' support and the

level of numerical attributes correlating to that support.

RELOADED optimizes the efficiency of LOADED by replacing the covariance matrix with a series of classifiers. These two algorithms are highly efficient when used in their intended manner, but their detection accuracy may be increased further. Both LOADED and RELOADED are supervised techniques, which means that training datasets are required.

Mixed-type data can also be analyzed by combining various single-type approaches. Koufakou et al. [46][48] suggest ODMAD for high-dimensional datasets, which separates outlier detection for categorical and computational fields. Outliers in categorical fields are identified by numbering, while outliers in numerical fields are identified depending on the data's distance from the numerical field's core. While this approach is simple, it ignores the relationships between categorical and numerical fields. Additionally, it takes a strong grasp of the instance space in order to feed in multiple user-defined thresholds for outlier filtering.

Tran and Jin [47] model symbolic features with a $C4.5$ decision tree and numerical fields with a Gaussian mixture model (GMM), with deviations found by adding the weighted sum of the decision tree and GMM scores to a predetermined threshold. As with ODMAD, this approach involves thorough hyper-parameter fine-tuning to determine the optimum weights for all scores and an appropriate threshold for outliers.

Ye et al. [49] took a different approach, detecting top-k deviations using a forecast outlier identification system (PODM) that takes into account both discrete and continuous fields. When finding deviations, the underlying theory is that the appearance of an anomalous instance in a lower dimension projection would be abnormally lower than the average. Thus, all continuous fields are converted to discrete values, and the data space is partitioned into several cells. A Gini entropy and an outlying degree are calculated on a series of subspaces onto which all instances are projected to determine whether a single subspace is an exception. Finally, outliers are detected in the anomalous subspaces using low-density cells. The primary disadvantage of this approach is the need to discretize numerical values using a unit length, that is, an equal-width interval. Due to the often wide variety of numerical attributes involved, all of these fields must be carefully preprocessed.

The closest literature to our proposed framework was presented by Zhang and Jin [50]. In their work, the idea of patterns found in the majority of data in terms of their attributes was adapted, with the number of patterns equal to the number of categorical data attributes. A "pattern" is defined in this case using a linear logistic regression with numerical explanatory variables and a single

categorical response variable. The advantage of a regression-based model is that it elucidates the functional relationship between the attributes. [51]. While this method models certain relationships among the attributes, the logistic regression it applied is highly susceptible to anomalous instances.

In the statistic research community, regression models have been extensively explored in the robust statistics sub-domain. For instance, various linear regression methods with enhanced robustness for numerical-only data have been proposed in previous research [52][53]. Liu [54] also proposed a robust variant of logistic regression and demonstrated its ability to accommodate outliers.

Incorporating an input-output relationship and a Gaussian latent variable, we present a model in this work that uses a rigorous statistical approach to capture attribute dependencies. Through combining robust architecture and simplified linear models, our new solution is able to process mixed-type data while retaining its anomaly resilience. The new architecture seeks to achieve a high level of detection precision while delivering the outputs in a reasonable amount of time. The process identifies deviations directly from the input dataset without the need for a training data collection.

2.3 Model Design

Subsection 2.3.1 starts by formalizing the problem. In Subsection 2.3.2, we explore how to model mixed-attributes using simplified linear models with an error buffering variable to account for the effects caused by the abnormal instances. Subsection 2.3.3 presents the integrated Bayesian hierarchical model.

2.3.1 Problem Formulation

Consider a dataset $S = \{s_1 \cdots s_N\}$ with P response (or dependent) variables $\{y_1(s) \cdots y_P(s)\}$ and D explanatory (or independent) variables $\{x_1(s) \cdots x_D(s)\}$. The distinction between response (for instance, the listing price of a house) and explanatory (for instance, the square-footage and the number of rooms of the house) variables is determined through the domain-specific prior-knowledge of the user, and both variables may be considered response (dependent) variables in certain cases. The dependent variables can be a variety of data types, including numerical, binary, and/or categorical variables, while the explanatory attributes are usually numerical. The aim is to

model the data distribution and distinguish instances of abnormal response variables or explanatory attributes.

Types of Anomalies: According to the categorization of the attributes, anomalies may also be introduced as abnormal response variables or peculiar explanatory attributes. Thus, we classify anomalies into three categories depending on their source attribute groups:

1. **Type I Anomalies** are caused by anomalies appear in the response variables.
2. **Type II Anomalies** are caused by anomalies appear in the explanatory attributes.
3. **Type III Anomalies** are caused by anomalies appear in both response variables and explanatory attributes.

An anomaly is defined by any entity that possesses attributes that cause it to behave as if it belonged to any of the preceding three groups. Anomalous objects usually deviate significantly from the data's patterns and can thus be identified by our statistical model.

Predictive Process: The first step makes use of numerical response variables, which are often thought to have a Gaussian distribution. As a result, the Gaussian predictive method can be used in this case. The following regression formulation captures the instances' behavior:

$$Y(s) = X(s)\beta + \omega(s) + \varepsilon(s) \tag{2.1}$$

This formula indicates that similar instances should have similar explanatory attributes. The regression effect β is a $P \times D$ matrix that represents the weights of the explanatory attributes with regard to the response variables. The dependency effect $\omega(s)$ is a Gaussian process used to capture the correlation between the response variables and a local adjustment provided for each response attribute. The error effect $\varepsilon(s)$ quantifies the difference between the real behavior and the expected normal behavior. In this model, the instances are assumed to be independent and identically distributed (*i.i.d.*), which can be formulated by the Gaussian likelihood

$$\pi(Y(s)|\eta(s)) \sim \mathcal{N}(Y(s)|\eta(s), \sigma^2_{num}), \tag{2.2}$$

where $\eta(s) = X(s)\beta + \omega(s) + \varepsilon(s)$, and σ^2_{num} is set to a small number to allow the random effects for ω and ε to be captured.

2.3.2 GLM and Robust Error Buffering

GLM (Generalized Linear Model) is based on the assumption that non-numerical data are generated using a specific distribution in the exponential family. Taking the example of the binary response type, each of the response variables is based on an assumption of a Bernoulli distribution, which follows $\pi(Y(s)|\eta(s)) \sim Bernoulli(g(\eta(s)))$, where g is a logit link function that transforms the numerical likelihood value to the success probability denoted by the Bernoulli distribution. Here, a sigmoid function is utilized for the transformation, for example, $g(x) = \frac{1}{1+exp(-x)}$. The GLM is capable of handling a variety of data types, including binary, count, multinomial, categorical, and others. In this work, we consider four types of data attributes, namely numerical, binary, count, and categorical. The specific use of GLM in our modeling approach will be discussed in subsection 2.3.3.

The robust error buffer is a key element of the proposed new algorithm. A hidden random variable is used to absorb the influence of measurement error, noise, or abnormal behavior. This mechanism's objective is to distinguish between predicted normal behavior and errors. Instead of a standard Gaussian distribution, the error variance $varepsilon$ is modeled using a Student-t distribution. The Student-t distribution is a special distribution that has a heavier tail than a Gaussian distribution. It is widely applied in robust statistics [36]. The heaviness of its tail can be controlled by the degrees of freedom hyper-parameter. In the case that the degrees of freedom reaches infinity, the Student-t distribution is equivalent to a Gaussian distribution. The probability density function of the Student-t distribution $st(0, df, \sigma)$ can be formulated by

$$p(\varepsilon) = \frac{\Gamma(\frac{df+1}{2})}{\Gamma(df/2)}(\frac{1}{\pi df\sigma})^{\frac{1}{2}}(1 + \frac{\varepsilon^2}{df\sigma})^{-\frac{df}{2}-\frac{1}{2}}, \tag{2.3}$$

where df is the degrees of freedom, Γ represents the gamma function, and σ represents the scale parameter. The proposed model considers the error effect $varepsilon(s)$ to be Student-t process at mean 0, and having a predetermined degree of freedom and a diagonal covariance matrix. Two advantages of using this error buffer in the model can be described as follows. First, the robustness of parameter estimation can be improved, and the normal behavior is more correctly inferred.

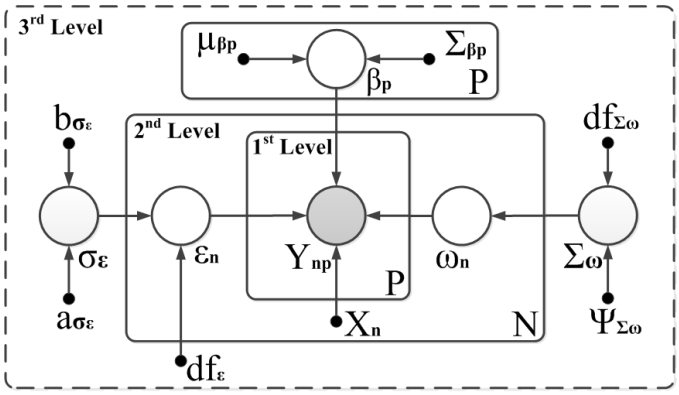


Figure 2.1: Graphical Model Representation

Table 2.1: GLM Information

Type	Likelihood	Link Function
Numerical	Gaussian	Identity
Binary	Bernoulli	Logit Function
Count	Poisson	Log Function
Categorical	Nominal	Logit Function

Second, because this hidden variable absorbs irregular effects, it is possible to find deviations by examining the values of the variables.

2.3.3 A Bayesian Hierarchical Model

By combining the components discussed in the preceding paragraphs, we can complete the design of the new algorithm. Figure 2.1 presents the graphical representation of the proposed model.

The proposed model is built on a Bayesian hierarchical model that allows for automatic parameter learning while enabling users to assign values to hyper-parameters based on their prior domain knowledge.

The first (observation) level of this hierarchical model captures the relationships among the pairs of response variables. This level of model associates to the predictive process. The GLM captures the dependencies between the response variables and the hidden effects. Four different types of data are taken into account here, namely numerical, binary, count, and categorical. Each of the data types corresponds to a distinct category of probability. The proposed method models these data types

using the standard GLM settings, which assumes Gaussian, Bernoulli, and Poisson distributions for integer, binary, and count data types, respectively. We employ the modeling strategy defined in [55] for categorical results, which composes the a categorical variable into a one-hot vector. The categorical response variable is generalized to include K binary variables, where K denotes the variable's category count. Table 2.1 describes the GLM probability and relation function associated with each data form.

The second (latent variable) level is consisted of the latent variables. This level includes latent elements corresponding to the correlation effect among the response variables and the error buffer, namely $omega$ and $varepsilon$. This level is primarily concerned with modeling the relationships between latent variables and parameters. To be more precise, we may construct the following equations:

$$\omega(s) \sim \mathcal{N}(\omega(s)|0, \Sigma_\omega), \tag{2.4}$$

$$\varepsilon(s) \sim St(\varepsilon(s)|0, \sigma_\varepsilon, df), \tag{2.5}$$

where σ_ε denotes a diagonal covariance matrix that indicates the variances of the error effects, Σ_ω is the covariance matrix used to model the covariance between the response variables, and df is the degree of freedom hyper-parameter. For simplicity, we will refer to the latent variable collection as nu.

The third (parameter) level specifies the regression coefficients and the corresponding conjugate priors of the model parameters, as well as the covariance matrix for ω, and the covariance matrix for ε, denoted by Σ_ω and σ_ε, respectively.

We formulate the prior distribution of the regression coefficients β by

$$\beta_p \sim \mathcal{N}(\beta|\mu_{\beta p}, \Sigma_{\beta p}), \tag{2.6}$$

where β_p is the regression coefficient associated with the p-th response variable, and $\mu_{\beta p}$ and $\Sigma_{\beta p}$ are the Gaussian distribution's hyper-parameters of each β_p.

In order to minimize the dimensionality of θ, we preserve only the variance of ω and ε in each

response variable, as well as their correlation:

$$\sigma^2_{\varepsilon p} \sim IG(a_{\varepsilon p}, b_{\varepsilon p}), \tag{2.7}$$

$$\Sigma_\omega \sim IW(\Phi, df_\omega), \tag{2.8}$$

Each response variable's variance $\sigma^2_{\varepsilon p}$ is assigned an inverse gamma distribution, while the covariance matrix of ω is assigned an inverse Wishart distribution. The symbols $a_{\varepsilon p}$, $b_{\varepsilon p}$, Φ ,and df_ω denote the prior distributions' hyper-parameters. Now that the model is well developed, the next step is to adapt it to the dataset. The following section introduces the system for detecting anomalies and describes the procedure for approximating Bayesian inference for the model.

2.4 Framework and Inference

This section discusses the framework for the anomaly detection method, mathematical inference for the model, computing costs, and optimization strategies. This thesis introduces two new architectures, one based on the INLA approach and the other on the EP-EM algorithm.

2.4.1 INLA Framework

To begin, we propose an approach based on INLA [56], a relatively recent technique for Bayesian inference approximation. By taking the mode as the mean and the second-order derivative at the mode as the variance, the Laplace approximation (LA) process approximates an arbitrary distribution to Gaussian (or covariance matrix in multivariate distribution). The core concept behind INLA is to iteratively estimate the marginal posteriors of latent variables using the Laplace approximation. The advantage is that the INLA fitting method is especially successful when the model parameters are located in a lower-dimensional space.

Framework

Algorithm 1 introduces the MITRE-INLA architecture, which is made up of three main components: Laplace approximation, latent variable estimation, and anomaly detection.

Phase 1: Laplace Approximation. Steps 1 to 10 demonstrate how the INLA method is constructed using two Laplace approximations in a nested structure. A maximum *a posteriori* (MAP) is performed to θ in the outer loop. Because we can formulate the posterior distribution of θ in the following form:

$$p(\theta|Y) \propto \frac{p(\nu, Y, \theta)}{p(\nu|Y, \theta)}, \tag{2.9}$$

and our objective is to maximize $p(\theta|Y)$, we can now consider the posterior density function $p(\theta|Y)$ as an objective function, which transforms the situation into an optimization problem. The next step is to update this objective function $p(\theta|Y)$ with values for each input. Thus, the inner loop (Steps 4–6) is executed to obtain the Laplace approximation to $p(\nu|Y, \theta)$. By utilizing a Taylor expansion on $p(\nu|Y, \theta)$, an analytical formulation can be achieved that restructures this density function into the quadratic form:

$$p(\nu|Y, \theta) = -\frac{1}{2}\nu^T Q\nu + \nu^T b. \tag{2.10}$$

Then, at Step 5, the latent variable set ν can be updated with $\nu = Q - 1b$ for each iteration. ν will converge to a local optimum after a few iterations. This process of updating, known as iterative reweighted least-squares (IRLS) [57], usually converges after less than five iterations. Steps 7–9 determine the value of the objective function $p(\theta|Y)$ at the local optimum $\hat{\nu}$, and change θ accordingly. The process continues until θ is converged.

Phase 2: Variable Estimation. After obtaining the mode of $p(\theta|Y)$ (denoted by $\hat{\theta}$), samples from the neighbors of $\hat{\theta}$ in the space of θ can be collected and utilized to infer the optimum values of θ and ν. This process is similar to the importance sampling [58] method that is widely used for numerical analysis, the difference being that samples are only taken from the mode area in the space. Steps 11–19 illustrate this procedure.

Phase 3: Anomaly Detection. Finally, Steps 20–28 demonstrate the mechanism used to identify outliers. After determining the optimal ν, which is denoted by $\nu*$, we can conduct anomaly detection using the optimized latent variable collection. We begin by extracting and inspecting the fitted error buffer $\varepsilon*$ from $\nu*$. Step 24 describes the method for detecting deviations in terms of a predefined threshold. This threshold is usually set to three times the standard deviation; that is, the absolute Z-score equals three, similar to how Gaussian distribution deviations are labeled.

Computational Cost and Optimization

The computing cost of statistical modeling techniques is typically a factor to consider; if the approach is to be used online, performance speed becomes much more critical. The technique here is to approximate complex computations, sacrificing a small amount of precision in exchange for a large improvement in performance. As stated in the Experimental Results section, these optimizations have been successfully tested experimentally.

Latent Computational Optimization: As shown in Algorithm 1, Step 5 is a major bottleneck in the framework. Due to the high dimensionality of the latent variable set, the matrix inversion calculation is extremely sluggish. To optimize this step, ν is divided into $\varepsilon, \omega, and \beta$ and then updated iteratively, as in the Gibbs sampling process. Algorithm 2 illustrates the approximation process. The first nine steps demonstrate how the original procedure is decomposed into three smaller phases. In Steps 2, 5, and 8, the algorithm updates the latent variables in the same sense as the original. Each of the three latent variables is expanded separately and then introduced into the Gaussian quadratic form in Equation (2.10) and updated using IRLS, that is, iteratively performing $\beta = Q_{\beta}^{-1} b_{\beta}, \varepsilon = Q_{\varepsilon}^{-1} b_{\varepsilon}$, and $\omega = Q_{\omega}^{-1} b_{\omega}$. Each time $update\_\nu$ is invoked, two variables are set, and one is modified, significantly reducing the computational cost. The complexity of the original INLA update is $O((P(2N + D))3)$, which is proportional to the scale of the latent variable set for the matrix inversion, while the complexity of the optimized update is reduced to $O(N3)$.

Approximate Parameter Estimation: Another bottleneck we identified in Algorithm 1 is that when the dimension of the parameter space is large, sampling and evaluating the weight from the $\hat{\theta}$ neighborhood is computationally expensive. As a result, we approximate the optimum in Step 11 by reducing the size of the samples. Although the estimated parameters will not perfectly fit the optimum, the latent variable set will still follow approximately a similar pattern if the estimated parameters are close to the optimum. Our experience shows that the approximated $\hat{\theta}$ is usually close enough to the optimal solution of θ. Due to the fact that the anomaly detection system only uses the latent variables, a small bias in the parameter estimation would have little effect on the detection outcome.

2.4.2 EP Framework

While the INLA approach will provide a reasonable approximation to the inference, the grid integration scheme in the neighborhood of θ (Line 11 in Algorithm 1) induces a substantial decrease to the efficiency when the dimension of θ increases [59]. As an alternative, we developed another approximate Bayesian inference method for the MITRE model using expectation propagation (EP) in conjunction with the VEM paradigm. EP has been shown to provide more accurate outcomes than Laplace's method in terms of predictive distributions and marginal likelihood estimations [37].

Framework

Under this context, we can use EP to approximate the inference of the latent variables. The assumption is rooted in an expectation-maximization loop based on mean-field theory in order to approximate the optimum model parameters θ. Algorithm 3 demonstrates the process flow of MITRE-EP.

Phase 1: Approximate Inference. The first ten steps demonstrate how to do approximate inference using EP-EM, which will be discussed in greater detail in the following subsection. The inner loop employs the EP algorithm to estimate latent variables, while the outer loop utilizes the EM algorithm to estimate the parameter set θ. More details of this framework are discussed in the next subsection.

Phase 2: Anomaly Detection. This process follows the INLA framework procedures. We remove the fitted error buffer $\varepsilon*$ from the approximate $\nu*$ and inspect it for anomalies. Steps 15–17 describe how deviations are observed in relation to a predefined threshold. This threshold is usually set to three times the standard deviation; that is, the Z-score equals three, similar to how Gaussian distribution deviations are identified.

Approximate Inference

In the EM algorithm, the E-step estimates the expectation of the posterior distribution $p(\theta|Y)$. The posterior is proportional to the joint distribution according to Bayes' theorem.

$$p(\theta|Y) \quad \propto \quad p(Y|\theta)p(\theta) \tag{2.11}$$

Thus, the expectation of log posterior of the complete-data for a general θ value is given by

$$\mathcal{Q}(\theta, \theta^{old}) = \mathbb{E}_{\nu}[\ln p(\nu, Y|\theta)|\theta^{old}] + \ln p(\theta) + Const$$

where θ^{old} is the parameter values of the current iteration. $Const$ denotes the constant that does not depend on θ. In the M-step of EM, the new parameter estimation θ^{new} is estimated by maximizing the expectation $\mathcal{Q}$, such that

$$\theta^{new} = \arg\max_{\theta} \mathcal{Q}(\theta, \theta^{old}) \tag{2.12}$$

Now, the first goal is to estimate the expectation $\mathbb{E}_{\nu}[\ln p(\nu, Y|\theta)|\theta^{old}]$. As discussed previously, the mixed-type GLM and the Student-t prior create a complicated joint distribution, making inference intractable. As a result, EP is used to estimate the inference in this case.

When starting with the inner EP process, we first expand $p(\nu, Y|\theta)$:

$$p(\nu, Y|\theta) = p(Y|\nu, \theta)p(\nu|\theta) \tag{2.13}$$

As shown in the model structure (Fig 2.1), the likelihood component can be formulated into a product form:

$$p(Y|\nu, \theta) = \prod_{n=1}^{N} \prod_{p=1}^{P} p(y_{np}|\beta_p, \omega_{np}, \varepsilon_{np}) \tag{2.14}$$

and

$$p(\nu|\theta) = \prod_{n=1}^{N} p(\omega_n|\theta) \prod_{p=1}^{P} p(\varepsilon_{np}|\theta) \tag{2.15}$$

The joint distribution thus becomes

$$p(\nu, Y|\theta) = \prod_{n=1}^{N} p(\omega_n|\theta) \prod_{p=1}^{P} p(\varepsilon_{np}|\theta)p(y|\beta_p, \omega_{np}, \varepsilon_{np})$$

With the application of EP, $p(\nu, Y|\theta)$, the intractable distribution, can be approximated by a Gaussian

distribution, and the approximated Gaussian is as following:

$$\begin{aligned} q(\nu) &= \frac{1}{Z} q_0(\nu) \prod_{n=1}^{N} \prod_{p=1}^{P} q_{np}(\nu) \\ &= \mathcal{N}(\nu|h, C) \end{aligned} \tag{2.16}$$

where $q_0(\nu) = \prod_{n=1}^{N} p(\omega|\theta)$ is the prior. Each $q_n(\nu)$ approximates the product of the likelihood and the Student-t prior according to the n-th entity, so that

$$q_{np}(\nu) = \mathcal{N}(\nu|m_{np}, R_{np}) \approx p(\varepsilon_{np}|\theta)p(y|\beta_p, \omega_{np}, \varepsilon_{np})$$

Given that GLM is integrated to the model, the above equation yields $p(y_n|\beta, \omega_n, \varepsilon_n)$, with different forms for various data types. To be more specific, each instance n can contain P answer variables of various types. We use the subscript p to denote the index of an instance's response variables. For instance, the likelihood for numerical data is assumed to be a Gaussian distribution

$$p(y_{np}|\beta_p, \omega_{np}, \varepsilon_{np}) = \mathcal{N}(y_{np}|x_n\beta_p + \omega_{np} + \varepsilon_{np}, \lambda) \tag{2.17}$$

To condense the equations, we denote $x_n\beta_p + \omega_{np} + \varepsilon_{np}$ by $\eta_{np}\nu$, where η_{np} is a $(2 + D) * P$ vector. The EP algorithm iterates over all elements with respect to n and p. In each round, the algorithm updates the approximated distribution using the deletion and inclusion strategy described in Minka's work [37]. The procedure is repeated until m and R converge:

1. **Deletion:** remove q_{np} from the full proposal distribution q

$$q^{\backslash n,p}(\nu) = \mathcal{N}(\nu|h^{\backslash n,p}, C^{\backslash n,p}) \tag{2.18}$$

where

$$h^{\backslash n,p} = h + C^{\backslash n,p} R_{np}^{-1}(h - m_{np}) \tag{2.19}$$

$$C^{\backslash n,p} = (C^{-1} - R_{np}^{-1})^{-1} \tag{2.20}$$

2. **Moment Matching:** finding the updated approximated proposal distribution $q^{(new)} \sim$

$\mathcal{N}(h^{(new)}, C^{(new)})$ by moment matching

$$q_{prop}(\nu) = q^{\setminus n,p}(\nu)p(\varepsilon_{np}|\theta)p(y|\beta_p, \omega_{np}, \varepsilon_{np})$$

The mean and variance of $q^{\setminus n,p}$ can be calculated by IRLS. Beginning with a Taylor expansion series to the combined distribution at the point ν_0,

$$\begin{aligned} q_{prop}(\nu) &= q_{prop}(\nu_0) + \nabla_\nu q_{prop}(\nu_0)(\nu - \nu_0) \\ &\quad + \frac{1}{2}\nabla\nabla_\nu q_{prop}(\nu_0)(\nu - \nu_0) \end{aligned} \tag{2.21}$$

We can then rearrange the series into squared form

$$q_{prop}(\nu) = -\frac{1}{2}\nu^T Q\nu + b\nu + const$$

such that reaching a local optimum at $Q^{-1}b$. By matching the coefficients from the preceding equations,

$$\begin{aligned} b(\nu_0) &= \nabla\nabla_\nu q_{prop}(\nu_0)\nu_0 - \nabla_\nu q_{prop}(\nu_0) \\ Q(\nu_0) &= \nabla\nabla_\nu q_{prop}(\nu_0) \end{aligned}$$

The mode of q_{prop} can be found by IRLS iteratively

$$\nu^{(i+1)} = Q^{-1}(\nu^{(i)})b(\nu^{(i)}) \tag{2.22}$$

in each iteration, given a starting point $\nu^{(0)}$.

Given that different probability functions are inferred for different data types, this step is done differently depending on the data type.

The gradient and Hessian of the numerical likelihood can be formulated by:

$$\begin{aligned}
\nabla_\nu q_{prop}(\nu) &= \frac{-\eta_{np}}{\lambda}(\eta_{np}^T\nu - Y_{np}) \\
&- C^{\backslash n,p^{-1}}(\nu - h^{\backslash n,p}) \\
&- \frac{(df+1)\varepsilon_{np}}{df\sigma_\varepsilon + \varepsilon_{np}^2}\eta_{\varepsilon_{np}} \\
\nabla\nabla_\nu q_{prop}(\nu) &= \frac{-1}{\lambda}\eta_{np}\eta_{np}^T - C^{\backslash n,p^{-1}} \\
&- \frac{(df+1)(\sigma_\varepsilon df - \varepsilon_{np}^2)}{(\sigma_\varepsilon df + \varepsilon_{np}^2)^2}\eta_{\varepsilon_{np}}\eta_{\varepsilon_{np}}^T
\end{aligned}$$

3. **Update**: Each approximated distribution is updated by

$$q_{np}(\nu) = \frac{q^{(new)}}{q^{\backslash n,p}} \tag{2.23}$$

From Equation (2.23) and the definition above we have

$$\begin{aligned}
R_{np}^{-1} &= C^{(new)^{-1}} - C^{\backslash n,p^{-1}} \\
m_{np} &= R_{np}(C^{(new)^{-1}}h^{(new)} - C^{\backslash n,p^{-1}}h^{\backslash n,p})
\end{aligned} \tag{2.24}$$

After approximating the assumption for the latent variable ν, the first component of the expectation can be expressed as the following formula:

$$\begin{aligned}
&\mathbb{E}_\nu[\ln p(\nu, Y|\theta)|\theta^{old}] \\
= &\sum_{n=1}^{N}\ln\mathcal{N}(\hat{\omega}_n|0,\Sigma_\omega) + \sum_{n=1}^{N}\sum_{p=1}^{P}\ln\mathcal{ST}(\hat{\varepsilon}_{np}|0,\sigma_{\varepsilon_p})
\end{aligned}$$

where $\hat{\nu}$ is the expected value of ν.

The expectation of the log distribution function θ is:

$$\begin{aligned}
& \mathcal{Q}(\theta, \theta^{old}) \\
= \; & \mathbb{E}_{\nu}[\ln p(\nu, Y|\theta)|\theta^{old}] + \ln p(\theta) \\
= \; & \sum_{n=1}^{N}\sum_{p=1}^{P} \ln p(y_{np}|\beta_p, \hat{\omega}_{np}, \hat{\varepsilon}_{np}) + \sum_{p=1}^{P} \ln \mathcal{N}(\beta_p|\mu_{\beta p}, \Sigma_{\beta p}) \\
+ \; & \sum_{n=1}^{N} \ln \mathcal{N}(\hat{\omega}_n|0, \Sigma_{\omega}) + \sum_{n=1}^{N}\sum_{p=1}^{P} \ln \mathcal{ST}(\hat{\varepsilon}_{np}|0, \sigma_{\varepsilon_p}) \\
+ \; & \ln IW(\Sigma_{\omega}|\Phi, df_{\omega}) + \sum_{p=1}^{P} \ln IG(\sigma_{\varepsilon_p}|a_{\varepsilon_p}, b_{\varepsilon_p})
\end{aligned} \tag{2.25}$$

To make a clearer statement, we separate $\mathcal{Q}(\theta, \theta^{old})$ into β, ω, and ε.

$$\begin{aligned}
& \mathcal{Q}^{(\beta)}(\theta, \theta^{old}) \\
& \qquad = \sum_{p=1}^{P}\sum_{n=1}^{N} \ln p(Y_{np}|X_n\beta_p + \hat{\omega}_{np} + \hat{\varepsilon}_{np}) + \sum_{p=1}^{P} \ln p(\beta_p)
\end{aligned}$$

$$\begin{aligned}
& \mathcal{Q}^{(\omega)}(\theta, \theta^{old}) \\
& = \sum_{n=1}^{N} \left(-\ln 2\pi - \frac{1}{2}\ln|\Sigma_{\omega}| - \frac{1}{2}\hat{\omega}_n^T \Sigma_{\omega} \hat{\omega}_n \right) \\
& + \frac{df_{\omega}}{2}|\Phi| - df_{\omega}\ln 2 - \ln \Gamma_2(\frac{df_{\omega}}{2}) \\
& \qquad - \frac{df_{\omega}+3}{2}\ln|\Sigma_{\omega}| - \frac{1}{2} tr(\Phi \Sigma_{\omega}^{-1})
\end{aligned}$$

$$
\begin{aligned}
&\mathcal{Q}^{(\varepsilon)}(\theta, \theta^{old}) \\
&= \sum_{n=1}^{N}\sum_{p=1}^{P}\left(\ln\Gamma(\frac{df+1}{2}) - \ln\Gamma(\frac{df}{2}) - \frac{1}{2}\ln\pi df\right. \\
&\left. -\frac{1}{2}\ln\sigma^2_{\varepsilon_p} + \frac{df+1}{2}\ln\left(1+\frac{\hat{\varepsilon}^2_{np}}{2\sigma^2_{\varepsilon_p}}\right)\right) \\
&\qquad + \sum_{p=1}^{P}\left(a\ln b - \ln\Gamma(a) - (a+1)\ln\sigma^2_{\varepsilon_p} - \frac{b}{\sigma^2_{\varepsilon_p}}\right)
\end{aligned}
$$

In the M-Step, we maximize the objective by calculating the root of $\mathcal{Q}(\theta, \theta^{old})$. Because this is indeed an intractable situation, IRLS is used to find an approximate solution. The root of θ can be approximated by iteratively updating the value and inputting the gradient and Hessian from Equation 2.22.

For $\sigma^2_{\varepsilon p}$, the gradient and Hessian are:

$$
\begin{aligned}
&\frac{\partial}{\partial\sigma^2_{\varepsilon p}}\mathcal{Q}^{(\varepsilon)}(\theta, \theta^{old}) \\
&= -\frac{N}{2\sigma^2_{\varepsilon p}} + (\frac{df+1}{2})\sum_{n=1}^{N}\left(\frac{-\hat{\varepsilon}^2_{np}}{4\sigma^2_{\varepsilon p} - 2\hat{\varepsilon}^2_{np}}\right) \\
&\qquad + \left(\frac{-(a+1)}{\sigma^2_{\varepsilon p}} + \frac{2b}{(\sigma^2_{\varepsilon p})^2}\right) \qquad (2.26)
\end{aligned}
$$

$$
\begin{aligned}
&\frac{\partial^2}{\partial^2\sigma^2_{\varepsilon p}}\mathcal{Q}^{(\varepsilon)}(\theta, \theta^{old}) \\
&= \frac{2N}{4(\sigma^2_{\varepsilon p})^2} + (\frac{df+1}{2})\sum_{n=1}^{N}\left(\frac{4\sigma^2_{\varepsilon p}\hat{\varepsilon}^2_{np} - (\hat{\varepsilon}^2_{np})^2}{(4\sigma^2_{\varepsilon p} - 2\hat{\varepsilon}^2_{np})^2}\right) \\
&\qquad + \frac{a+1}{(\sigma^2_{\varepsilon p})^2} - \frac{4b}{(\sigma^2_{\varepsilon p})^3} \qquad (2.27)
\end{aligned}
$$

Since β corresponds to distinct likelihood functions for distinct data types, the maximization can be computed separately for each type. The calculations presented here are only for numerical data. For other data types, β can be approximated using a Laplace approximation (see supplemental material).

The root can be found by setting the first-order derivation to zero for the β_p corresponding to the numerical data type. Therefore, each β_p can be updated with the following:

$$\begin{aligned}\beta_p^{(new)} &= (\frac{1}{\lambda}\sum_{n=1}^{N} X_n^T X_n + \Sigma_{\beta_p}^{-1})^{-1} \\ &\quad \times (\frac{1}{\lambda}\sum_{n=1}^{N} X_n^T (Y_{np} - \hat{\omega}_{np} - \hat{\varepsilon}_{np}) + \Sigma_{\beta_p}^{-1}\mu_{\beta_p}) \quad (2.28)\end{aligned}$$

Computational Optimizations

Several optimization schemes are suggested in this subsection to further improve the framework's performance. The approach here is to approximate these complicated computations, sacrificing some precision for a substantial improvement in performance. Additionally, as defined in the Experimental Results section, these optimizations have been successfully tested experimentally.

Correlation Parameter Reduction: Given that process complexity is proportional to the dimension of the parameters, one way to minimize complexity is to reduce the number of parameters. We used the mutual information [60] approach to measure the scores of the dependencies between each pair of answer attributes for this optimization. Through specifying a user-defined parameter K, only the top K attribute correlations need to be considered for fitting. The correlation parameter is reduced from $\binom{P}{2}$ to K using this approximation. When P is large, this approximation greatly reduces the computational cost.

Sub-sampling Fitting: When the data set is large, we can further reduce the complexity by sampling a subset of the data and then detecting anomalies using the model constructed from those samples. The same level of accuracy is achieved where the scale of the sampled instances and the number of sample batches are sufficient. This optimization holds true for the INLA-based architecture as well.

2.5 Experiments

Comprehensive MITRE experiments were performed to assess the following performance measures: accuracy detection, time efficiency, and effect of parameters. This section shows the outcomes of the experimental analyses and organizes them as follows: The benchmark methods are presented in Section 2.5.1. Section 2.5.2 addresses the precision of identification, time efficiency and influence of parameters using synthetic datasets, and 2.5.3 Section includes a thorough assessment of the efficacy of MITRE as used in real-world datasets. Findings of these analyses are addressed in Section 2.5.4. Both the experimental sets have been carried out with a 2.4 GHz Intel Dual Core CPU and 8GB RAM on a Windows 7 computer.

2.5.1 Benchmark Approaches

Eight benchmark approaches were compared: LOADED [17], RELOADED [18], KNN-CT, LOF-CECT, OCS-PCT, OCS-RBF, FB-LOF [61], and ODMAD [46]. LOADED, RELOADED, and ODMAD are methods for detecting mixed-type anomalies; OCS-RBF (One-class SVM with RBF kernel) and FB-LOF (feature bagging with a LOF base) are methods for detecting numerical sort anomalies. We preprocessed the OCS-RBF and FB-LOF datasets by transforming categorical fields to their binary representations and normalizing all fields using the min-max method. The remaining three methods are both combined single-type anomaly detection methods composed of six single-type anomaly detection methods, including three numerical anomaly detection methods (KNN, LOF, and OCS (one-class SVM) [28]) and three categorical anomaly detection methods (CT, CECT, and PCT, both from [62]). Das and Schneider [62] demonstrated that these methods outperformed other categorical methods, which led to their inclusion as the benchmark methods for categorical attributes in this report. The integrated methods carried out the identification procedures independently and then normalized the scores to create a single metric. For both LOADED and RELOADED, the best results for each dataset reported here are based on the true anomaly labels. We utilized the popular settings of the model parameters (correlation threshold = [0.1, 0.2, 0.3, 0.5, 0.8, 1]; frequency threshold = [0, 10, 20]; τ = [1,2,3,5]). For ODMAD, the $minsup$ value is set to be the reciprocal of the number of categories of each categorical field. The parameters were chosen based on 10-fold cross validations for the remaining benchmark methods.

2.5.2 Synthetic Study

For the synthetic data analysis, we evaluated the proposed frameworks' detection precision, compared their time costs to those of benchmark approaches, and studied the effect of parameter settings.

Datasets

The synthetic data were generated by the following process:

$$Z(s) = X(s)\beta + \omega(s) \tag{2.29}$$

First, we generated explanatory attributes X ($N \times D$) from a normal distribution, with a set of $\beta : D \times P$ and the covariance matrix between the attributes for a set of $Z = [Z_1, \ldots, Z_P]$. We converted the Z_i to different types for each element, such that

$$Y(s) = g_{type}(Z(s)), \tag{2.30}$$

where g is the link function for the given type, such as binary or count. Next, the anomalies were injected by shifting the values of $Y(s)$ randomly. For example, for binary data, the value is shifted by swapping the classes of the observations. The following experiments produced a variety of synthetic datasets based on the test's objective. Each dataset was created with the aim of containing between 8% and 10% of anomalies.

Detection Accuracy

We evaluated model inference accuracy on four sets of synthetic data containing various combinations of data types, namely SynNB, SynNC, SynBC, and SynNBC. The symbols N, B, and C, respectively, denote numerical, binary, and categorical data forms. As shown in Table 2.2, the detection accuracy was compared through synthetic datasets. MITRE-EP and MITRE-INLA outperformed the other benchmark approaches substantially since these simulated datasets had quite a strong input-output relationship. Although a synthetic analysis is not often persuasive due to the presumptions used to generate the data, these findings show unequivocally that when the input-

Table 2.2: Detection Rate Comparison Among Synthetic Datasets (Precision, Recall)

Dataset	MITRE-EP	MITRE-INLA	KNN-CT	LOF-CECT	OCS-PCT	RELOADED	LOADED	OCS-RBF	FB-LOF	ODMAD
SynNB	**1.00**, 0.69	**1.00, 0.89**	0.11, 0.11	0.25, 0.50	0.29, 0.56	0.29, 0.56	0.28, 0.56	0.72, 0.72	0.06, 0.06	0.08, 0.61
SynNC	0.89, 0.82	**1.00, 0.89**	0.06, 0.06	0.40, 0.33	0.28, 0.56	0.29, 0.56	0.27, 0.56	0.72, 0.72	0.33, 0.33	0.06, 0.50
SynBC	**0.89, 0.67**	0.71, **0.67**	0.06, 0.06	0.33, 0.17	0.20, 0.39	0.20, 0.39	0.03, 0.06	0.33, 0.33	0.11, 0.11	0.08, 0.50
SynNBC	**0.92**, 0.73	0.80, **0.77**	0.04, 0.04	0.75, 0.33	0.33, 0.63	0.32, 0.59	0.21, 0.41	0.59, 0.59	0.04, 0.04	0.12, 0.58

Table 2.3: Time Cost Comparison in Terms of Size of N (seconds)

Method \ Size	300	500	1K	10K	100K	1M	2M
MITRE-EP	2.74	4.39	8.72	97.58	113.43	1662.17	>7200
MITRE-INLA	1.99	8.38	32.65	133.54	>7200	>7200	>7200
KNN-CT	0.01	0.02	0.07	5.80	313.28	>7200	>7200
LOF-CECT	0.01	0.03	0.11	29.31	N/A	N/A	N/A
OCS-PCT	0.02	0.03	0.12	12.11	N/A	N/A	N/A
RELOADED	0.01	0.14	0.19	0.44	4.48	87.90	258.77
LOADED	0.07	0.10	0.22	2.54	23.59	249.13	484.02
OCS-RBF	0.02	0.03	0.07	8.38	606.84	>7200	>7200
FB-LOF	0.05	0.13	0.29	14.66	708.66	>7200	>7200
ODMAD	0.01	0.01	0.03	0.24	3.39	46.80	169.50

Experiments that exceeded the available memory resources are denoted by N/A

Experiments that ran over 2 hours are considered a failure

Table 2.4: Time Cost Comparison in Terms of Size of P (seconds)

Method \ Size	10	25	50	100	200	300
MITRE-EP	8.77	19.06	153.30	1020.43	9344.83	>7200
MITRE-INLA	42.05	319.37	>7200	>7200	>7200	>7200
KNN-CT	0.14	158.24	>7200	>7200	>7200	>7200
LOF-CECT	6.69	596.32	>7200	>7200	>7200	>7200
OCS-PCT	0.24	>7200	>7200	>7200	>7200	>7200
RELOADED	0.24	0.46	1.02	2.26	5.44	7.86
LOADED	1.16	60.83	>7200	>7200	>7200	>7200
OCS-RBF	0.01	0.01	0.01	0.01	0.02	0.02
FB-LOF	0.30	0.37	0.51	0.87	1.63	2.11
ODMAD	0.018	>7200	>7200	>7200	>7200	>7200

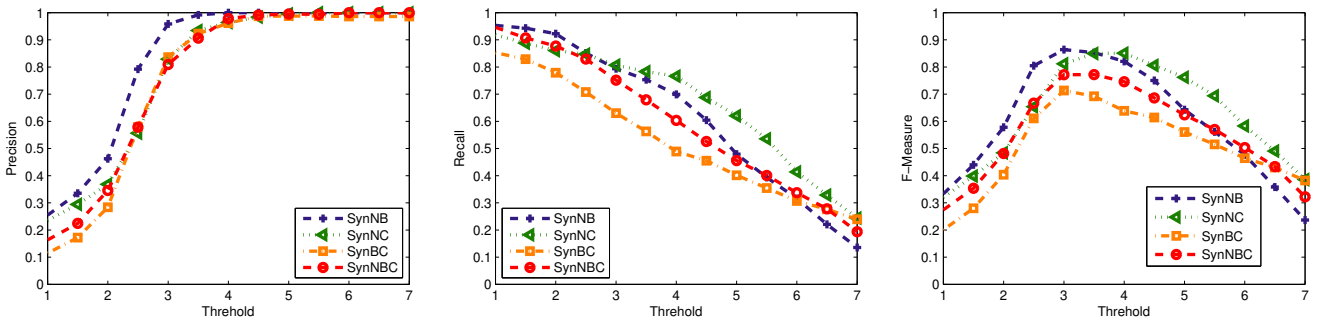
Experiments that ran over 2 hours are considered a failure

output relationships are good, and the dataset has access to pre-knowledge, MITRE significantly outperformed all the benchmark methods evaluated.

Time Cost

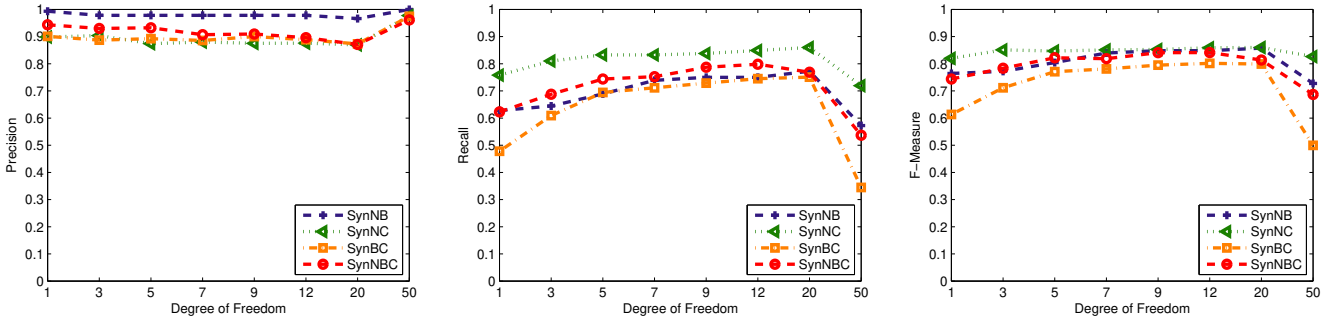
The time costs incurred by MITRE and the benchmark approaches were compared in this series of experiments. These studies were performed on simulated datasets in which the normal instances were created using a GLM model for mixed-type attributes, and the anomalous instances were generated using random shifting. Table 2.3 compares the time costs of different approaches for datasets of varying instance sizes. Experiments that last more than two hours are deemed invalid.

Although our solution required more time than the benchmark approaches, it achieved significantly higher detection precision in a comparable amount of time, as described in Section 2.5.2 and the subsequent experimental findings for real-world evidence. Table 2.4 illustrates the time usage as the size of P increases. The majority of benchmark approaches were unable to accommodate data of higher dimensions. For example, due to ODMAD's exhaustive searching system, the computational expense increased exponentially with the number of categorical fields. MITRE-INLA often suffered from a large dimension of P scale owing to the method of θ estimation (Section 2.4.1 Phase 2). MITRE-EP demonstrated its ability to complete the process with a P size of 100 within an hour.



(a) Precision (b) Recall (c) F-measure

Figure 2.2: Impact of Error Threshold



(a) Precision (b) Recall (c) F-measure

Figure 2.3: Impact of Degree of Freedom

Impact of Parameters

The threshold for detecting deviations and the degrees of freedom of the Student-t prior are two critical user feedback parameters in our current system architecture. The selection of these two criteria does have an impact on results. We performed this series of experiments using the previously mentioned synthetic datasets, namely SynNB, SynNC, SynBC, and SynNBC, for different instance sizes.

Threshold of Anomalies: The aim of this series of experiments was to analyze the influence of the threshold used to identify anomalies. As previously mentioned, we used SynNB, SynNC, SynBC, and SynNBC, with N equal to 100, 300, 500, and 1000. Ten variant realizations were produced for each type-size combination. The threshold was evaluated in 0.5 intervals from 1 to 7. Figure 2.2 compares the influence of various accuracy, recall, and F-measure thresholds. Figure 2.2 (a) demonstrates that all datasets adopt a similar trend, with accuracy rising dramatically from 1 to 3.5 and then decreasing to a modest level after 3.5. When the threshold was raised to 5, all datasets achieved their highest level of precision. Figure 2.2 (b) demonstrates that recall decreased steadily over all data forms. The F-measure shows a more clear trend in Fig. 2.2 (c). Here, for all data forms, the peaks fell between the 3 and 4 thresholds, confirming our theory about the threshold setting.

Degree of Freedom: This series of tests examined the effect of the proposed model's degree of freedom df. We repeated the previous set of experiments using the same synthetic data sets. When evaluating the degree of freedom impact, the error threshold was set to three. As shown in Figure 2.3, setting a lower df usually results in higher accuracy, as the absorbed error emphasizes the difference between anomalies and normal instances. A small improvement in the F-measure from df=1 to df=3 has a noticeable effect; the inference was unable to converge to the global optimum in a finite number of iterations when df was set to a value less than 3.

2.5.3 Real-life Data Study

Datasets

We validated our method using 14 real-world datasets, available in the UCI machine learning repository [63]. Table 2.5 contains basic details for each of these datasets. The forms are denoted in the table by the letters N, B, and C, which stand for numerical, binary, and categorical, respectively.

Table 2.5: Information in Real Datasets

Dataset	***Instances***	***Attrs***	***Type***	***Response***
Abalone	4177	9	C, N	1, 9
Yeast	1324	9	C, N	1, 6
WineQuality	4898	12	C, N	1, 12
Heart	163	11	C, B	3, 6
Autompg	398	8	C, N	1, 8
Wine	178	13	C, N	1, 2
ILPD	583	10	B, N	1, 2
Blood	748	5	B, N	4, 5
Concrete	103	10	B, N	8, 9, 10
Parkinsons	197	23	B, N	2, 18
Pima	768	8	B, N	3, 9
KEGG	53414	23	B, N	5, 7, 12, 13
MagicGamma	19020	11	B, N	1, 2, 11
Census	299285	42	C, N	6, 19, 25, 42

In our experiment, we used the response fields displayed in Table 2.5 as Y and the remaining attributes as X; the number refers to the nth column of the raw dataset.

Anomaly Labels

Because the above datasets do not provide true anomaly labels, we preprocessed the data to obtain true anomaly labels in two different ways:

1. Rare Classes. We found rare categorical groups in the first category of datasets (Autompg, Abalone, WineQuality, Yeast, and Heart). These rare class instances were defined as true anomalies using the same technique as previously published anomaly detection studies [64, 65].

2. Random Shifting. For the remaining datasets, we treated all data artifacts as normal and produced anomalies using the standard contamination technique defined in [32] and [34]. We choose 10% instances at random and shifted the values in random fields. We moved the numerical values by 2.5 standard deviations for numerical attributes and changed the binary and categorical values to alternate values for binary and categorical attributes. Every dataset's data was preprocessed with a total of twenty separate artificial anomaly combinations, and the average of the twenty findings was computed for each test.

Table 2.6: Detection Rate Comparison Among Real Datasets (Precision, Recall, F-measure, AUC)

Dataset	MITRE-EP	MITRE-INLA	KNN-CT	LOF-CECT	OCS-PCT
Abalone	**0.78**, 0.29, **0.42**, 0.94	0.25, **0.62**, 0.36, 0.98	0.16, 0.33, 0.22, 0.69	0.02, 0.04, 0.03, 0.49	0.20, 0.42, 0.27, 0.67
Yeast	**1.00**, 0.47, **0.64**, **1.00**	0.55, 0.67, 0.60, 0.59	0.29, 0.57, 0.38, 0.28	0.05, 0.10, 0.07, 0.15	0.21, 0.44, 0.28, 0.62
WineQuality	**0.50**, 0.29, 0.36, 0.93	0.33, 0.65, **0.44**, 0.95	0.03, 0.06, 0.04, 0.03	0.02, 0.04, 0.03, 0.03	0.04, 0.07, 0.05, 0.09
Heart	**1.00**, 0.57, 0.72, **0.99**	0.95, 0.75, 0.84, 0.98	0.46, **0.76**, 0.57, 0.50	0.45, 0.75, 0.56, 0.50	0.24, 0.43, 0.31, 0.50
Autompg	**0.47**, **1.00**, **0.64**, **1.00**	**0.47**, **1.00**, **0.64**, 0.99	0.00, 0.00, 0.00, 0.00	0.00, 0.00, 0.00, 0.00	**0.47**, **1.00**, **0.64**, 0.99
Wine	0.22, 0.66, **0.33**, 0.67	**0.33**, 0.30, 0.31, 0.63	0.09, 0.17, 0.12, 0.50	0.09, 0.17, 0.12, 0.50	0.09, 0.18, 0.12, 0.51
ILPD	0.22, **0.70**, 0.33, **0.78**	**0.84**, 0.18, 0.30, 0.77	0.26, 0.49, **0.34**, 0.60	0.12, 0.23, 0.16, 0.57	0.25, 0.49, 0.33, 0.59
Blood	**0.70**, 0.35, **0.47**, 0.74	0.56, 0.15, 0.24, **0.82**	0.23, 0.44, 0.30, 0.37	0.08, 0.15, 0.10, 0.35	0.24, 0.48, 0.32, 0.57
Concrete	0.57, **0.84**, **0.68**, **0.95**	**0.79**, 0.59, **0.68**, 0.92	0.07, 0.13, 0.09, 0.51	0.07, 0.14, 0.09, 0.50	0.09, 0.40, 0.15, 0.52
Parkinsons	0.60, **0.74**, **0.67**, **0.94**	**0.78**, 0.46, 0.58, 0.91	0.21, 0.42, 0.28, 0.37	0.23, 0.44, 0.30, 0.38	0.21, 0.41, 0.28, 0.50
Pima	0.79, **0.55**, **0.65**, 0.78	**0.83**, 0.27, 0.40, **0.82**	0.25, 0.48, 0.33, 0.44	0.06, 0.11, 0.08, 0.40	0.25, 0.49, 0.33, 0.66
KEGG	0.87, **0.65**, **0.77**, 0.75	0.59, 0.41, 0.48, 0.53	0.24, 0.46, 0.31, 0.37	N/A	N/A
MagicGamma	**0.67**, **0.66**, **0.66**, **0.83**	0.60, 0.55, 0.57, 0.82	0.14, 0.28, 0.19, 0.45	N/A	N/A
Census	**0.60**, **0.71**, **0.65**, **0.81**	0.51, 0.58, 0.54, 0.71	N/A	N/A	N/A

Dataset	RELOADED	LOADED	OCS-RBF	FB-LOF	ODMAD
Abalone	0.00, 0.00, 0.00, 0.29	0.00, 0.00, 0.00, 0.00, 0.50	0.25, 0.25, 0.25, **0.99**	0.04, 0.04, 0.04, 0.74	0.01, 0.62, 0.02, 0.58
Yeast	0.00, 0.00, 0.00, 0.35	0.66, 0.66, 0.66, 0.58	0.63, 0.63, 0.63, 0.96	0.21, 0.21, 0.21, 0.50	0.05, **0.88**, 0.09, 0.91
WineQuality	0.00, 0.00, 0.00, 0.43	0.12, 0.12, 0.12, 0.51	0.11, 0.11, 0.11, 0.81	0.19, 0.19, 0.19, 0.75	0.12, **1.00**, 0.21, 0.91
Heart	0.51, 0.51, 0.51, 0.89	1.00, 0.16, 0.28, 0.72	0.65, 0.65, 0.65, 0.89	0.35, 0.35, 0.35, 0.57	0.99, **0.99**, **0.99**, **0.99**
Autompg	0.29, 0.29, 0.29, 0.70	0.33, 0.57, 0.42, 0.74	0.57, 0.57, 0.57, 0.98	0.10, 0.10, 0.10, 0.85	0.04, **1.00**, 0.08, 0.99
Wine	0.17, 0.36, 0.23, 0.59	0.12, 0.12, 0.12, 0.50	0.24, 0.24, 0.24, **0.77**	0.16, 0.16, 0.16, 0.60	0.10, **0.70**, 0.17, 0.56
ILPD	0.00, 0.00, 0.00, 0.50	0.09, 0.09, 0.09, 0.50	0.23, 0.23, 0.23, 0.68	0.09, 0.09, 0.09, 0.50	0.14, 0.71, 0.23, 0.45
Blood	0.03, 0.01, 0.02, 0.51	0.09, 0.09, 0.09, 0.50	0.39, 0.39, 0.39, 0.79	0.14, 0.14, 0.14, 0.58	0.19, **0.52**, 0.28, 0.64
Concrete	0.13, 0.26, 0.17, 0.58	0.08, 0.08, 0.08, 0.50	0.32, 0.32, 0.32, 0.72	0.17, 0.17, 0.17, 0.59	0.08, 0.43, 0.13, 0.49
Parkinsons	0.29, 0.21, 0.24, 0.59	0.07, 0.07, 0.07, 0.50	0.14, 0.14, 0.14, 0.72	0.21, 0.21, 0.21, 0.60	0.18, 0.53, 0.27, 0.65
Pima	0.10, 0.28, 0.15, 0.59	0.05, 0.05, 0.05, 0.50	0.52, 0.52, 0.52, 0.78	0.07, 0.07, 0.07, 0.52	0.31, 0.28, 0.29, 0.51
KEGG	0.78, 0.26, 0.39, 0.61	0.10, 0.10, 0.10, 0.50	0.51, 0.51, 0.51, **0.95**	N/A	**0.98**, 0.26, 0.41, 0.63
MagicGamma	0.28, 0.02, 0.04, 0.56	0.10, 0.10, 0.10, 0.50	0.29, 0.29, 0.29, 0.81	N/A	0.12, 0.46, 0.19, 0.55
Census	0.27, 0.30, 0.28, 0.64	0.10, 0.10, 0.10, 0.50	N/A	N/A	0.33, 0.29, 0.31, 0.61

Experiments that the methods failed to process are denoted by N/A

Table 2.7: Performance (AUC) Comparison for Various Random Shift Values

	MITRE-EP					MITRE-INLA				
Dataset \ **Shift**	1.0	1.5	2.0	2.5	3.0	1.0	1.5	2.0	2.5	3.0
Wine	0.59	0.58	0.63	0.67	0.67	0.62	0.61	0.59	0.63	0.64
ILPD	0.66	0.69	0.75	0.78	0.80	0.66	0.68	0.71	0.77	0.75
Blood	0.68	0.77	0.75	0.74	0.87	0.69	0.77	0.80	0.82	0.84
Concrete	0.74	0.84	0.93	0.95	0.97	0.81	0.82	0.93	0.92	0.97
Parkinsons	0.79	0.87	0.91	0.94	0.94	0.79	0.89	0.88	0.91	0.91
Pima	0.71	0.79	0.82	0.78	0.89	0.67	0.72	0.73	0.82	0.78
KEGG	0.60	0.59	0.67	0.75	0.74	0.54	0.55	0.53	0.53	0.68
MagicGamma	0.76	0.77	0.80	0.83	0.84	0.73	0.74	0.80	0.82	0.82
Census	0.70	0.71	0.74	0.81	0.79	0.69	0.68	0.72	0.71	0.76

Detection Accuracy

The primary objective of these observations on real-world datasets was to test our proposed approach for detecting anomalies. Table 2.6 measures the accuracy, recall, F-measure, and area under the curve (AUC) metrics associated with a variety of different methods. The findings indicate that MITRE outperformed the benchmark methods in terms of average precision and memory, indicating that the majority of instances detected by MITRE as exceptions were true positives. Additionally, MITRE obtained the highest average AUC, indicating that our anomalous score metric consistently provided the highest detection rate.

Although many other benchmark methods obtained a strong AUC, they often had a high incidence of false positives or negatives. For instance, ODMAD obtained an AUC of greater than 0.9 on the *Yeast*, *WineQuality*, and *Auto-mpg* datasets, with near-perfect recalls, but its precision did not surpass 0.12 due to the predicted threshold for anomalous scores being set too low, resulting in several normal instances being incorrectly identified as anomalies. RELOADED, LOADED, and ODMAD all needed the input of several parameters as a collection of hard thresholds, which had a major impact on their output. These methods have the potential to work well after any parameter tuning, but they would almost certainly fail in many realistic situations where the size and basis are uncertain. In comparison, the proposed new form, MITRE, applies the absolute value of the Z-score as the anomalous score and a statistical exclusion threshold under the Gaussian assumption, which is commonly used in a wide variety of real-world situations. Regardless of the measure used to describe the various data properties, this ranking reflects the predictive importance of the data

and the degree to which it deviates from normal activity on a normalized basis.

Additionally, the findings show MITRE's efficacy on massive real-world datasets such as *Census*. The sub-sampling fitting strategy (discussed in Section 2.4.2) significantly reduced the computational expense while retaining a high detection rate. In comparison, due to computational storage and time constraints, the benchmark methods LOF-CECT and OCS-PCT were unable to process any of the datasets containing a large number of instances (*KEGG*, *MagicGamma*, and *Census*) because they exceeded the available memory resources; KNN-CT, OCS-RBF, and FB-LOF have encountered difficulties with these large datasets, with running times exceeding two hours.

Table 2.7 illustrates the effect of outlier importance on random shift data sets. We contrasted the AUCs of importance ranges for spontaneous shifting ranging from one standard deviation to three standard deviations. In general, values that were changed 1.5 times the standard deviation or less were difficult to detect, although our methods worked well in some situations even when the deviations were not dramatically shifted. According to our observations of datasets of mixed-type results, where the changing degree had little effect on numerical attributes, binary and categorical deviations were both appropriately observed, and the anomalous scores for these instances were correctly ranked.

2.5.4 Result Analysis

The findings of the preceding experiments show that MITRE-EP is a highly accurate and reliable technique for identifying irregularities in mixed-type data sets. It outperforms the other benchmark approaches by approximately 10%–30% when compared to KNN-CT, LOF-CECT, OCS-PCT, OCS-RBF, and ODMAD, and 20%–40% when compared to LOADED, RELOADED, and FB-LOF. Three major discoveries were confirmed by the empirical results.

1) Efficient Approximation Process: The suggested approximate inference schemes provide identification results more efficiently and accurately. MITRE-EP outperforms the INLA-based approach in terms of computational performance and outlier detection precision on a wider variety of real-world data sets.

2) Effectiveness on Large Mixed-type Datasets: When processing more complex data sets, such as *Census*, *KEGG*, and *MagicGamma*, LOF-CECT and OCS-PCT were unable to complete the process due to substantial memory consumption. Due to their high time complexity, KNN-CT,

OCS-RBF, and FB-LOF struggled on the *Census* dataset. Our alternative approaches were able to complete the process in a comparable amount of time without experiencing power constraints.

3) Input-Output Relationship: When the datasets have clear input-output relationships for the explanatory properties of the response variables, the MITRE methods outperform the benchmark methods in terms of detection precision. Notably, we made these distinctions using the relationships suggested by the dataset providers for the majority of real-world datasets.

2.6 Conclusions

This study introduces an innovative unsupervised method for general-purpose anomaly detection on mixed-type datasets. The new approach employs EP and VEM to combine multivariate statistical process models with approximate Bayesian inference. The predictive model is composed of generalized linear models and latent variables with robust error buffering. Approximation and optimization techniques allow more precise and efficient inference for the proposed predictive process model. Experiments on synthetic and real-world datasets showed conclusively that our proposed anomaly detection system outperformed existing frameworks in terms of detection accuracy.

Algorithm 1 MITRE-INLA

Require: The response variables Y and explanatory attributes X
Ensure: The anomalous instances

set $\theta = \theta_0$
while $\theta \neq argmax_\theta(p(\theta|Y))$ **do**
  set$\nu = \nu_0$
  while $\nu \neq argmax_\nu(p(\nu|Y,\theta))$ **do**
    $\nu = update\_\nu$
  end while
  $\hat{\nu} = \nu$
  $L = likelihoodofp(\theta|Y,\hat{\nu})$
  $\theta = update\_\theta(L)$
end while
θ_{sample} = sample from neighborhood of θ
set $L_{\theta_{samples}}, \hat{\nu}_{\theta_{sample}} = \phi$
for all θ_s in $\theta_{samples}$ **do**
  $\hat{\nu}_{\theta_s} = argmax_\nu(p(\nu|Y,\theta_s))$
  $L_{\theta_s} = likelihoodofp(\theta|Y,\hat{\nu}_{\theta,s})$
  put $\hat{\nu}_{\theta_s}$ into $\hat{\nu}_{\theta_{samples}}$
  put L_{θ_s} into $L_{\theta_{samples}}$
end for
$weight = normalize(L_{\theta_{samples}})$
$\nu^* = \hat{\nu}_{\theta_{samples}} * weight$
$\varepsilon^* = getErrorBuffer(\nu^*)$
set $AnomalySet = \phi$
for all ε_s^* in ε^* **do**
  if $\varepsilon_s^* > ErrorThreshold$ **then**
    put s in $AnomalySet$
  end if
end for
return $AnomalySet$

Algorithm 2 $update\_\nu$

Require: The original latent variable $\varepsilon, \omega, \beta$
Ensure: The updated latent variable $\varepsilon_{new}, \omega_{new}, \beta_{new}$
while $\beta \neq argmax_{\beta}(p(\beta|Y, \theta, \varepsilon, \omega))$ **do**
  $\beta = update\_\beta(\varepsilon, \omega)$
end while
while $\varepsilon \neq argmax_{\varepsilon}(p(\varepsilon|Y, \theta, \beta, \omega))$ **do**
  $\varepsilon = update\_\varepsilon(\beta, \omega)$
end while
while $\omega \neq argmax_{\omega}(p(\omega|Y, \theta, \varepsilon, \beta))$ **do**
  $\omega = update\_\omega(\varepsilon, \beta)$
end while
return $\varepsilon_{new} = \varepsilon, \omega_{new} = \omega, \beta_{new} = \beta$

Algorithm 3 MITRE-EP

Require: The response variables Y and explanatory attributes X
Ensure: The anomalous instances
set $\theta = \theta_0$
while $\theta \neq argmax_{\theta}(p(\theta|Y))$ **do**
  set $\nu = \nu_0$
  while $\nu \neq argmax_{\nu}(p(\nu|Y, \theta))$ **do**
    $\nu = update\_\nu$
  end while
  $\hat{\nu} = \nu$
  $L = likelihoodofp(\theta|Y, \hat{\nu})$
  $\theta = update\_\theta(L)$
end while
set $\nu^* = mode(p(\nu|Y, \theta))$
$\varepsilon^* = getErrorBuffer(\nu^*)$
set $AnomalySet = \phi$
for all ε_s^* in ε^* **do**
  if $\varepsilon_s^* > ErrorThreshold$ **then**
    put s in $AnomalySet$
  end if
end for
return $AnomalySet$

Chapter 3

Automatic Categorical Anomaly Detection in Music Genre Datasets

Outlier detection, alternatively referred to as anomaly detection, is a critical subject that has been researched over the decades. An effective anomaly detection system is capable of identifying outliers in a dataset and is capable of improving data integrity through the removal of identified anomalies. The algorithms have been applied effectively to various types of data in a variety of fields, including cyber security, transportation and financial domains. However, the area of music information retrieval (MIR) has a dearth of related studies. We implement and test various state-of-the-art outlier detection techniques designed for music datasets in this work. More precisely, we present an experimental analysis of five outlier detection algorithms on two music genre recognition (MGR) datasets and propose an improved detection approach based on a robust categorical regression model. The evaluation results indicate that the new algorithm outperforms other approaches for outlier detection and is capable of locating problematic samples found by human experts. The proposed approach establishes a preliminary structure for detecting anomalies in music data that can be used in the future to increase data integrity. A portion of this work has been published in earlier articles [66][67].

3.1 Introduction

With the advancement of technology over the last few decades, enormous amounts of digital data are being produced. In order to accommodate for this rapid growth, efficiently and reliably leveraging its hidden knowledge became an active research area dubbed data mining.

Anomaly detection, which is one of the most extensively researched subjects in data mining and machine learning community, is a task that entails identifying outliers within the dataset under investigation. In general, an outlier refers to the instance that deviates from anticipated actions and should be highlighted. For instance, an outlier in a security monitoring system might be an intruder, while an outlier in bank transfer records can represent a fraudulent transaction.

Numerous approaches for identifying outliers in various forms of data have been proposed. Their effectiveness has been demonstrated in fields such as cyber security[68, 2004], finance[69], and transportation[70]. Outlier detection techniques may also be used to remove abnormal data instances during the pre-processing stage. Smith and Martinez[71] use a variety of anomaly identification methods to exclude deviations from the dataset, supplemented by many commonly used classification methods, to assess the performance improvements before and after the elimination of anomalous instances. The conclusion is that eliminating outliers will result in statistically meaningful increases in both the training quality and classification accuracy in the majority of cases.

Music datasets present a similar collection of difficulties for researchers in the MIR domain. In the work of Schedl et al., it is pointed out that several MIR experiments need distinct datasets and annotations based on the mission [72]. However, because music data annotation is a nuanced and subjective process, the accuracy of the annotated labels provided by professional human annotators differs significantly across datasets. This inaccuracy will introduce errors into the system, degrading its performance as a result.

One MIR challenge that has been identified as relevant to this problem is MGR. According to Sturm[19], the most commonly applied dataset for the MGR task is the GTZAN [7] dataset. A large number of existing systems assess their effectiveness using this dataset. However, in Sturm's research, it has been pointed out that this dataset does contain corrupted audio files, repeated music clips, and genre annotations that are incorrect. These problematic instances are unacceptable for training and evaluating an MGR application properly.

In this work, we propose an unsupervised method for addressing this issue and detecting anomalies in music datasets. The normal behavior of feature representations is captured using a statistics-based model. By leveraging a Student-t prior to the latent error variable in the model, the estimation of normal behavior remains robust and anomalous effects are absorbed into this latent variable. The following summarizes the contributions of this work:

- **An unsupervised music anomaly detection approach:** An unsupervised method for identifying anomalies in music datasets is proposed. There is no need for labeling anomalies.
- **A categorical anomaly detection model:** A regression-based categorical anomaly detection model is proposed, along with a robust estimation technique. Additionally, a method for approximating an analytically intractable inference is established.
- **An experimental study applying state-of-the-art outlier detection methods to MGR datasets:** A comprehensive experimental analysis is performed on two commonly used MGR datasets. Five existing methods for detecting outliers are implemented and applied to the datasets. The proposed solution outperforms current state-of-the-art approaches, as shown by the results.

The following is the organization of this work. In Section 3.2, the associated work of decting outliers in music data is summarized. Section 3.3 and Section 3.4 describe the application of feature extraction methods and the state-of-the-art outlier detection algorithms to MGR datasets. In Section 3.5, the proposed method is discussed, as well as the technique for approximation inference. Section 4.5 analyzes the findings of the experiments. Finally, Section 3.7 summarizes this work and discusses potential extension for future.

3.2 Related Work

In a high level, approaches for detecting outliers are commonly classified into five categories: (a) *distance-based* [26, 73], (b) *density-based* [74, 31], (c) *cluster-based* [38], (d) *classification-based* [28, 29], and (e) *statistics-based* [32, 33, 34, 39, 75, 76] methods.

The first group is the *Distance-based* methods. One classical work was proposed by Knorr et al. [26]. This approach computes distances between each pair of samples and identify outliers using

a distance threshold. Although the approaches in this group are typically straightforward and mostly efficient, their performance on accuracy usually suffers when the data instances are sparse or not distributed evenly. The fundamental concept was expanded by combining the idea of the distance criterion with the k-nearest neighbor (KNN)-based approach[73], which dynamically adjusts the distance threshold based on the distances of the k-nearest neighbors.

Density-based methods estimate the local densities surrounding each of the targeting points in order to identify anomalies. Different variations employ a variety of techniques for determining the local density, including the local outlier factor (LOF) [74] and the local correlation integral (LOCI) [31]. These methods are widely adopted and have been applied in various domains.

Clustering-based approaches, as proposed in [38], are based on a clustering algorithm first and then identifies the incorrectly clustered data instances as anomalies.

Classification-based approaches relies on an assumption that a classification algorithm that learns how to label anomalies. In this type of approaches, classification algorithms are used to differentiate the instances as normal instances and anomalies. Das et al. [28] demonstrate this using a one-class support vector machine (SVM) based approach, while Roth [29] presents one utilizing kernel Fisher discriminants.

The *Statistics-based* approaches are based on the assumption of that the data follow a particular underlying distribution so that anomalies can be detected by identifying instances with low probability according to the density distribution. Numerous works employ a variant of this principle, for instances, approaches based on the robust Mahalanobis distance [32], the minimum covariance determinant estimator [34], and the direction density ratio estimation [33]. One major challenge of these approaches is the masking and swamping effects: anomalies can distort the inference of the distribution parameters, resulting in skewed probability distributions. This effect could cause a skewed detector to classify normal instances as anomalies or vise versa. Recent developments have largely concentrated on the application of robust statistics to anomaly detection [39, 75, 76]. This problem is typically addressed by applying a robust estimation technique to ensure the pattern is not biased by outliers so that the normal behavior can be correctly captured.

Although the methods outlined above have been widely applied to various domains, the number of researches in the music domain is relatively limited. A novelty detection method for identifying anomalous instances was presented by Flexer et al. [77]. This method is able to detect the instances that have not been seen in the training data. The approach was demonstrated to be successful in a

cross-validation environment using an MGR dataset containing 22 genres. However, outliers are typically concealed in real-world datasets, and an outlier-free training dataset may not be accessible. Therefore, this suggested approach might be inapplicable to other music datasets. Hansen et al. [78] proposed a supervised method for automatically detecting anomalies based on parzen-window incorporating a kernel density estimation algorithm. This method was tested using a four-class MGR dataset containing audio data collected from radio stations. To reflect the music signals, a widely used collection of audio features called Mel-frequency cepstral coefficients (MFCCs) was extracted. However, this strategy has two inherent issues. First, no ground truth developed by human experts are provided in the assessment dataset. Second, although it is well established that MFCCs are useful for a variety of MIR tasks, they may not be sufficient to reflect music signals in outlier detection tasks.

Two approaches have been taken in this work to resolve these concerns. First, two widely used MGR datasets are introduced for evaluation. Sturm's study [19] has identified a set of anomalies that are repeated, distorted, and mislabeled music clips manually in the widely used GTZAN [7] dataset. With this analysis, a solid effectiveness validation for the outlier detection systems in the MGR dataset can be performed.

Furthermore, we expand the collection of music data descriptors. Along with the MFCCs, audio features that are frequently used in MIR tasks are also extracted to determine the compatibility with proven anomaly detection methods.

3.3 Feature Extraction

The feature extraction stage is critical because it converts the audio signal to a numerical vector representation that is suitable for subsequent data analysis. Tzanetakis and Cook presented three feature sets to classify any given music clip based on its timbral texture, rhythmic content, and pitch content[7] in an early analysis of automatic music genre classification. These features have demonstrated their effectiveness in classifying music genres and have been applied to a variety of music-related tasks. While numerous studies proposed more complicated features (for example, [79]) with improved classification performance on the GTZAN dataset, the original collection of features appears to be a reasonable solution for transforming music clips.

As a result, a collection of baseline features based on the features presented by Tzanetakis and Cook [7] is extracted to facilitate comparisons with the existing work. The extracted music features are grouped into spectral, temporal, and rhythmic. A block-wise analysis approach is used to extract these features. The audio signal is down-mixed to a mono signal as the first step. Following that, to obtain the time-frequency representation, a short-time Fourier transform (STFT) is then performed with a Hann window, a block size of 23 ms, and a hop size of 11 ms. Finally, various instantaneous characteristics are extracted from each block. The spectrum of each block is used to compute the spectral features. Temporal features are explicitly computed from each block's time domain signal. The rhythmic features are derived from the time domain signal's beat histogram. The extracted features are as follows (for more details of the implementations, refer to the work of Lerch [80]):

1. **Spectral Features** ($d = 16$): Spectral Centroid (SC), Spectral Roll-off (SR), Spectral Flux (SF), 13 Mel Frequency Cepstral Coefficients (MFCCs)
2. **Temporal Features** ($d = 1$): Zero Crossing Rate (ZCR)
3. **Rhythmic Features** ($d = 8$): Period0 (P0), Amplitude0 (A0), RatioPeriod1 (RP1), Amplitude1 (A1), RatioPeriod2 (RP2), Amplitude2 (A2), RatioPeriod3 (RP3), Amplitude3 (A3).

All extracted features are then aggregated into texture vectors according to the standard procedure as described in [7]; the current texture block has a length of 0.743 s. To construct a new feature vector, the mean and standard deviation of the feature vectors within this time span will be computed. Finally, the mean and standard deviation of all the texture blocks will be calculated, yielding a single feature vector to represent each individual recording in the dataset.

3.4 Outlier Detection Methods

3.4.1 Problem Definition

Given a collection of N music clips translated to a set of feature vectors $X = \{X_1, ..., X_N\}$ with the corresponding genre label $Y = \{Y_1, ..., Y_N\}$, where each Y_n is associated with one of the M

genres (i.e., $Y_n \in \{C_1, ..., C_M\}$), the objective is to determine the indices of the abnormal instances that have an incorrect label Y_n.

For this study, we compare the results of six well-known outlier detection methods from various categories implemented in Section 3.2 on an MGR dataset. The following sections detail the procedures:

3.4.2 Clustering

Clustering is as described in Sect. 3.2 a *cluster-based* approach. Our implementation of this approach clusters the data into ten groups using k-means. Based on the assumption that normal data is distributed near cluster centroids but outliers are not [81, 82], the outlierness score for a given instance can be defined as the difference between the point and the centroid of the majority within the same class.

3.4.3 KNN

The KNN method is a *distance-based* approach that usually determines each instance's anomalous score in terms of its instance to its k nearest neighbors [83]. It can be expressed as follows:

$$k\text{–}distance(P) = d(P, knn(P)) \tag{3.1}$$

where knn is the function that calculates the k-th nearest neighbor of a point P, and d is the function that returns the Euclidean distance between the given two points. We can compute the outlierness score by:

$$\frac{1}{k} \sum_{p \in neighbors_k(P)} k\text{–}distance(p) \tag{3.2}$$

A larger k usually refers to a more robust model against outliers. When k is small, the anomalous

score defined by this method is prone to be skewed by a small number of outliers. In our implementation of this approach in the experiment, we apply $k = 6$ in order to strike a balance between robustness and efficiency.

3.4.4 Local Outlier Factor

The local outlier factor (LOF)[74] is a *density-based* approach that augments the KNN method by computing the instances' local densities. It is one of the most widely used methods for detecting anomalies. It starts by defining the k-reachability distance:

$$k\text{–}reachDist(P, O) = max(k\text{–}distance(P), d(O, P)) \tag{3.3}$$

This formula represents the distance from O to P that is not less than the $k\text{–}distance$ of P. Here, the local reachability density is defined as the inverse of the average local rechability distances of a sample's k-nearest neighbors:

$$lrd(P) = 1/\left(\frac{\sum_{P_0 \in neighbors_k(P)} k\text{–}reachDist(P, P_0)}{|neighbors_k(P)|}\right) \tag{3.4}$$

Finally, the lof measures the average ratio of the k-nearing neighbors' local reachability densities to the point P:

$$lof(P) = \frac{\sum_{P_0 \in neighbors_k(P)} lrd(P_0)}{lrd(P)|neighbors_k(P)|} \tag{3.5}$$

In a densely distributed dataset, a point's mean distance to its neighbors will be shorter and vice versa. Because LOF applies a ratio as the anomalous score rather than the distance, it is convenient to detect outliers in clusters with varying densities.

3.4.5 One-Class SVM

The one-class SVM[84] is a one-class classifier-based approach that uses a binary classification model to identify outliers. Given a genre $m \in \{1, ..., M\}$, each sample in m can be classified as

in-class or out-of-class, with the outliers being the most probable. An SVM with a single class solves this quadratic programming problem:

$$\begin{aligned} \min_{w,\xi_i,\rho} \quad & \tfrac{1}{2}||w||^2 + \tfrac{1}{\nu N}\sum_i \xi_i - \rho \\ \text{subject to} \quad & (w\Phi(x_i)) \geq \rho - \xi_i, i = 1...N \\ & \xi_i \geq 0, i = 1...N \end{aligned} \tag{3.6}$$

where ν is a parameter that is defined as the upper bound fraction of outliers and a lower bound fraction of samples used as support vectors. w, ρ, and ξ are the parameters that define the separation hyperplane, and Φ is a function that projects the data instances into an inner product space. In such a way, the mapped data can be captured using certain kernels, for instance, a Gaussian radial basis function (RBF). After estimating the optimization of the objective function described above, a hyperplane is generated to denote the in-class instances from the anomalies. In our experiment, for each genre, we created a one-class SVM model, and each of the model classifies instances being labeled off-class as anomalies.

3.4.6 Robust PCA

Robust PCA[76] is a *statistics-based* method for decomposing a matrix X into the superposition of a low-rank matrix L_0 and a sparse matrix S_0, such that:

$$X = L_0 + S_0$$

The problem can be solved by the convex optimization of the principal component pursuit[76]:

$$minimize||L||_* + \lambda||S||_1 \tag{3.7}$$

$$\text{subject to } L + S = X \tag{3.8}$$

where $||L||_*$ is the nuclear norm of L, and λ is the sparsity constraint that determines how sparse S would be.

The matrix S_0 is a sparse matrix with the majority of entries being zero with a few non-zero

entries acting as outliers. In the experiment, we apply this approach to the music data and compute sparse matrices for each genre. Following that, we normalize the features using a standard Z-score normalization process and distinguish outliers by identifying instances with a maximum sparse matrix value that is three times greater than the unity standard deviation.

3.5 Robust Categorical Regression

We propose a novel *statistics-based* approach for identifying outliers in a categorical regression model.

3.5.1 Model

To begin with, we assume a linear input-output relation of Y and X, defined as:

$$g(Y) = X\beta + \varepsilon, \tag{3.9}$$

where g is the categorical link function, β is the regression coefficient matrix, and ε is a random variable that represents the white-noise vector of each instance. The link function g is a logit function paired with a category C_M; that is, $\ln\left(P(Y_n = C_m)/P(Y_n = C_M)\right) = X_n\beta_m + \varepsilon_{nm}$. Given that the probabilities of the categories add up to one, the following modeling equation can be deduced:

$$P(Y_n = C_m) = \frac{\exp\left\{X_n\beta_m + \varepsilon_{nm}\right\}}{1 + \sum_{l=1}^{M-1}\exp\left\{X_n\beta_l + \varepsilon_{nl}\right\}} \tag{3.10}$$

and

$$P(Y_n = C_M) = \frac{1}{1 + \sum_{l=1}^{M-1}\exp\left\{X_n\beta_l + \varepsilon_{nl}\right\}} \tag{3.11}$$

The coefficient vector β usually represents the decision boundary in a classification problem. In this approach, β is used to capture the normal behavior of the data.

Following the convention, we assume that each β_m obeys a Gaussian distribution with a predefined mean vector and a predefined covariance matrix Σ_β, i.e.,

$$\beta_m \sim N(\beta_m|\mathbf{0}, \Sigma_\beta) \tag{3.12}$$

Traditionally, the error factor ε is generally assumed to have a Gaussian distribution in regression applications. However, since the probability distribution is near zero at points far from the distribution mean, a Gaussian distribution lacks tolerance for anomalies. To increase the capability of capturing anomalies, it is suggested in the analysis of robust statistics to assume that this random variable has a heavy-tailed distribution[85].

In this work, we assume that the error is a zero-mean Student-t random variable, which has been shown to be an effective approach for increasing the robustness of the logistic regression model [54]. The probability density function is as the following:

$$p(\varepsilon|\sigma_\varepsilon^2, df) = \frac{\Gamma(\frac{df+1}{2})}{\Gamma(\frac{df}{2})\sqrt{\pi df \sigma_\varepsilon^2}} \left(1 + \frac{\varepsilon^2}{df\sigma_\varepsilon^2}\right)^{1(\frac{df+1}{2})} \tag{3.13}$$

where σ^2 is the scaling parameter, and df is the number of degrees of freedom. We leverage the Student-t variable as an "error buffer" in this case to absorb the error introduced by the anomaly instances, allowing us to distinguish easily between the anomalies and errors.

3.5.2 Approximate Inference

Due to the categorical nature of the answer, the inference becomes intractable. We introduce a Bayesian assumption into the model and two distinct methods for approximating Bayesian inference.

Variational Inference

The first approach is to use variational-EM algorithm [86] to approximate the inference. We start from the joint distribution of the model:

$$p(Y, \beta, \varepsilon) \propto p(Y|\beta, \varepsilon)p(\beta)p(\varepsilon) \tag{3.14}$$

Suppose there is a proposal distribution $q(Y, \varepsilon, \beta)$ that approximates p, such that $q \simeq p$. Based on the structure of the model, we can factorize q into two parts, i.e., $q(\varepsilon)$ and $q(\beta)$. By optimizing the optimal factors, the estimate of the variational variables is updated until the convergence criterion is satisfied. We use iterated re-weighted least squares (IRLS) to find the optimal value for each individual update. Applying the Taylor expansion to the log expectations above, we can obtain a quadratic form in $\ln q = -\frac{1}{2}\nu^T Q\nu + \nu^T b$, where ν represents the target variable to update, and

$$b(\nu) = \nabla\nabla_\nu q(\nu) - \nabla_\nu q(\nu) \tag{3.15}$$

$$Q(\nu) = \nabla\nabla_\nu q(\nu) \tag{3.16}$$

In each iteration, we update the value of ν by

$$\nu^{(new)} = Q^{-1}(\nu^{(old)})b(\nu^{(old)}) \tag{3.17}$$

Thus, by iteratively updating β and ε and estimating the gradient and Hessian in each iteration, the process will converge to a local optimum of these variables.

Markov Chain Monte Carlo

As an alternative approach, we apply the robust categorical regression model to a Markov chain Monte Carlo (MCMC)-based inference process. MCMC is a widely used method for approximating complex distributions that is based on iterative sampling from the target distribution. Numerous variants of MCMC have been proposed to address various types of problems; the most common of these variants are Gibbs sampling and Metropolis-Hasting sampling [87]. The benefit of MCMC is that when the sample size is sufficiently large, the approximated distribution approaches the target distribution very closely. In this work, a hybrid MCMC approach is applied to solve for the complicated conditional distribution introduced by the categorical regression and the Student-t prior.

In each iteration of the process, the latent variables are sampled from their full conditional distribution:

$$\begin{aligned}\log p(\beta_d|...) &= \log p(y|\beta_d, \beta^{\setminus d}, X, \varepsilon) + \log p(\beta_d) + C \\ &= \sum_{n=1}^{N} \Bigg(y_{nd} X \beta_d + \varepsilon_{nd} \\ &- \log\left(1 + \exp\left(X_n\beta_d + \varepsilon_{nd}\right) + \sum_{l\in\{1,...,D\}, l\neq d} \exp\left(X_n\beta_l + \varepsilon_{nl}\right)\right)\Bigg) + \log p(\beta_d) + C\end{aligned} \tag{3.18}$$

$$\begin{aligned}\log p(\varepsilon_{nd}|...) &= \log p(y|\varepsilon_{nd}, \beta, X, \varepsilon^{\setminus nd}) + \log p(\varepsilon_{nd}) + C \\ &= y_{nd} X_n \beta_d + \varepsilon_{nd} \\ &- \log\left(1 + \exp\left(X_n\beta_d + \varepsilon_{nd}\right) + \sum_{l\in\{1,...,D\}, l\neq d} \exp\left(X_n\beta_l + \varepsilon_{nl}\right)\right) + \log p(\varepsilon_{nd}) + C\end{aligned} \tag{3.19}$$

However, the distribution is complicated so it is not possible to sample from. We thus sample from proposal distributions $q(\beta_d)$ and $q(\varepsilon_{nd})$ for β_d and ε_{nd}, respectively.

A Metropolis-Hasting step is performed inside the Gibbs sampling to determine whether if the sample is adopted for this iteration. [88] Take β_d for example,

$$\begin{aligned}\text{draw } \tilde{\beta}_d &\sim q(\beta_d) \\ \text{draw } r &\sim uniform(0, 1)\end{aligned}$$

Then, the sampled value is accepted with a probability ρ, i.e.,

$$\beta_d^{(t)} = \begin{cases} \tilde{\beta}_d, & \text{if } r < \rho. \\ \beta_d^{(t-1)}, & \text{otherwise.} \end{cases} \tag{3.20}$$

where

$$\begin{aligned}\rho &= \frac{p(\tilde{\beta}_d|...)/q(\tilde{\beta}_d)}{p(\beta_d^{(t-1)}|...)/q(\beta_d^{(t-1)})} \\ &= \exp\left(\log p(\tilde{\beta}_d|...) - \log q(\tilde{\beta}_d) - \log p(\beta_d^{(t-1)}|...) + \log q(\beta_d^{(t-1)})\right) \end{aligned} \tag{3.21}$$

When the new sample is preferred, that is, it is less likely to be drawn from the proposal distribution but has greater value when plugged into the target pdf, the sample has a higher probability of being accepted.

The process will finally converge to a local optimum. The latent variable can be inferred by the mean of the samples without the burn-in samples.

3.6 Experiment

3.6.1 Experiment Setup

Several experiments on the well-known GTZAN dataset [7] are performed to test the state-of-the-art outlier detection methods mentioned in Sect. 3.4. This dataset contains ten music genres (i.e., blues, classical, country, disco, hip-hop, jazz, metal, pop, reggae, and rock), each with 100 audio tracks; each track is a 30-second sample from a complete mixture of music. Two sets of experiments are conducted for each method.

In the first set of experiments, we use a purified GTZAN dataset that is devoid of the glaringly misclassified and jittery music clips recorded in [19]. This configuration simulates the ideal scenario, in which the dataset is clean and all genres are well separated in the feature space. The results can be used to verify the validity of all the approaches. Two forms of injection experiments are conducted on this purified dataset: label injection and noise injection. The label injection process is accomplished by randomly selecting 5% of instances and swapping their genre labels to forge outliers. In this experiment, two sets of features are used to reflect the music data: the full feature set as defined in Section 3.3 and the baseline feature set using only 13 MFCCs as described in the work of Hansen et al. [78] for comparison. The noise injection process involves randomly selecting 5% of instances from the data and shifting 20% of their feature values by five times the standard deviation.

This experiment evaluates the methods' ability to detect manipulated data using the full feature set. We produce ten random realizations for each experiment and report the average evaluation results.

In the second set of experiments, we explicitly apply all the methods to the entire GTZAN dataset and compare the found outliers to the list of prominent genre labels and the obviously corrupted clips (*Hip-hop (38), Pop (37), Reggae (86)*) listed in [19]. This experiment simulates a real-world scenario in which outlier detection can identify outliers previously found by human experts.

All studies employ the same metrics for the performance measurements, which include the standard calculation of precision, recall, F-measure, and the area under the ROC curves (AUC).

3.6.2 Experiment Results

The results of the first set of experiments, which evaluated the methods' performance on detecting injected misclassification labels with full features, are shown in Table 3.1. With F-measures ranging from 0.1–0.57, the results lack reliability but are functional for some methods. The approach taken by *robust categorical regression* outperforms the others. Due to the explicit modeling of the input-output relationship between the features and the labels in RCR, it suits the data better than the other approaches. Surprisingly, simple methods such as *clustering* and *KNN* often work reasonably well in terms of AUC, outperforming more sophisticated approaches like *LOF* and *one-class SVM*. One possibility is that the label injection datasets contain outliers created by swapping dissimilar genres, such as changing the label from Jazz to Metal. As a result, *LOF* and *one-class SVM* decision boundaries might be biased towards the extreme values and perform poorly. On the other hand, the simple methods based on Euclidean distance, such as *clustering* and *KNN*, were able to distinguish these instances without becoming biased. In general, *statistics-based* approaches, such as the robust statistics-based methods *RPCA* and *robust categorical regression*, perform better on the label injection datasets.

Table 3.2 shows the results of the same experiments with only MFCCs as features. In general, the performance degrades precipitously. This result suggests that MFCCs might not be sufficiently representative for the outlier detection task.

Table 3.3 shows the results for the noise injection experiment. Methods based on density and distance methods, such as *CLUS*, *KNN*, and *LOF*, perform better at detecting corrected data. The primary difference between this type of outlier and others is that the abnormal behavior is directly

Method	Precision	Recall	F-measure	AUC
CLUS	0.23	0.23	0.23	0.74
KNN	0.26	0.26	0.26	0.77
LOF	0.11	0.11	0.11	0.57
SVM	0.06	0.32	0.10	0.52
RPCA	0.47	0.34	0.39	0.78
RCR	0.59	0.55	0.57	0.91

Table 3.1: Average Detection Performance Comparison of Label Injection with Full Features

Method	Precision	Recall	F-measure	AUC
CLUS	0.06	0.06	0.06	0.61
KNN	0.10	0.10	0.10	0.71
LOF	0.09	0.09	0.09	0.62
SVM	0.05	0.38	0.09	0.50
RPCA	0.30	0.20	0.24	0.65
RCR	0.52	0.40	0.45	0.87

Table 3.2: Average Detection Performance Comparison of Label Injection with MFCCs Only

shown in the feature space rather than being implicitly contained in the relationship between genre labels and the features. As a result, methods that directly detect outliers in the feature space usually outperform other methods such as *SVM*, *RCR*, and *RPCA*.

In the second set of experiments, we perform the anomaly detection on the complete GTZAN dataset with full features, with the aim of detecting the misclassified music clips reported by Sturm[19]. The experiment result is shown in Table 3.4. According to these metrics, none of these methods is capable of accurately detecting the anomalies. Although *SVM* and *RCR* have higher AUCs than the other approaches, their precision, recall and F-measures are still too poor to be useful in real-world scenarios. The *one-class SVM* method performs significantly better in this experiment than in the previous experiment. We hypothesized that when outliers are injected, the model is skewed by the

Method	Precision	Recall	F-measure	AUC
CLUS	0.92	0.90	0.91	0.99
KNN	0.99	0.98	0.99	1.00
LOF	1.00	0.98	0.99	1.00
SVM	0.05	0.41	0.09	0.50
RPCA	0.32	0.23	0.27	0.72
RCR	0.61	0.50	0.55	0.75

Table 3.3: Average Detection Performance Comparison of Noise Injection With Full Features

Method	Precision	Recall	F-measure	AUC
CLUS	0.15	0.13	0.14	0.54
KNN	0.18	0.15	0.16	0.56
LOF	0.18	0.15	0.16	0.59
SVM	0.09	0.63	0.15	0.66
RPCA	0.08	0.09	0.08	0.51
RCR	0.17	0.22	0.19	0.60

Table 3.4: Performance Comparison on GTZAN Detecting Sturm's Anomalies with Full Features

extreme values introduced by the outliers. However, in the real world, the differences between outliers and non-outliers are relatively subtle. As the *one-class SVM* in-class region is moderately expanded, it learns a more accurate decision boundary. As a result, it is more capable of identifying outliers.

It can be observed that both *statistics-based* approaches, *RPCA* and *RCR*, perform poorly compared to the previous experiment's performance. Due to the fact that these approaches are effective at capturing extreme values and preventing the model from being influenced by the outliers, they are relatively inefficient at differentiating subtle differences in the feature space. Therefore, the resulting performances are less than optimal.

3.6.3 Discussion

To further illustrate the relationship between different methods and outliers from various genres, we list the distribution of top 20 true and false outliers as determined by the anomalous scores of different methods, as well as the true distribution stated by Sturm [19]. The results are shown in Table 3.5. Interestingly, except the *one-class SVM*, the majority of methods identified most of the true outliers in *Disco* and *Reggae*. The top 20 of the *one-class SVM* contain 14 *Metal* outliers that are barely identified by the other methods. More specifically, the *one-class SVM* performed well in the *Metal* genre, with a precision of 14/26. Because most of the true outliers in the *Metal* genre can be classified as punk rock according to the definition on the online music library,[1] they could exhibit similar features with slight differences in the feature space and still be detected by the *One-Class SVM*. In *Reggae*, there is a jitter music clip that exhibits extreme values in the feature space, along with the other outliers. However, when the *One-Class SVM* is used with *Reggae*, only

[1] AllMusic: http://www.allmusic.com/

the jitter instance is captured, leaving the other outliers unaccounted for. These two observations demonstrate that *One-Class SVM* is particularly adept at separating outliers with subtle differences and is susceptible to becoming biased because of outliers with extreme values.

Four methods have about five *Jazz* instances in the top 20 false outliers. Although *Jazz* music has distinct characteristics and is easily distinguished by humans, it shows the highest average variance in terms of its features when compared to other genres. Thus, methods that compute Euclidean distances, such as *Clustering*, *KNN*, and *LOF*, as well as approaches that absorb variances as errors, such as *RPCA*, could report more false outliers in this scenario. Additionally, three methods contain approximately ten of the top 20 false outliers in *Pop*. This may be a result of the dataset's diversity of *Pop* music. For instance, while *Pop (12) - Aretha Franklin, Celine Dion, Mariah Carey, Shania Twain, and Gloria Estefan "You Make Me Feel Like A Natural Woman"* was not labeled as an outlier by the expert, it may be considered as *Soul* music. As a result of the *One-Class SVM* composition, this clip is also ranked at the top due to its anomalous score. Meanwhile, we observed that *RCR* contains 10 *Rock* false outliers in its top 20. This may be because it models the input-output relationship among all genres and thus allows the model to mix *Rock* with other partially overlapping genres, such as *Metal* and *Blues*.

To conclude, outlier identification of music data poses the following difficulties in comparison to other types of data. To begin, since the genre's definitions are vague, some of the tracks may be considered as both outliers and normal instances. This inconsistency can have an effect on both supervised and unsupervised approaches. Sturm's [89] work proposes a risk model to account for the lack of misclassification due to genre similarity. Second, music data are time-series in nature. In the experiment, we aggregate the sequences of feature matrix into a single feature vector in the current method because this allows for the immediate use of state-of-the-art methods. However, this approach ignores the temporal variations in the music signals and can omit critical knowledge needed to identify outliers with minor differences. Third, because the transformed low-level features could be unable to cover the high-level concept of music genre, it is nontrivial for the anomaly detection algorithms to identify the anomalies analyzed from the human experts. Finally, the data points are also distributed differently within each genre's feature space, and the outliers are spread unevenly throughout genres (e.g., *Metal* has 16 while *Blues* and *Classical* have none). A method or a particular parameter configuration may be able to perform well for some genres but poorly for the others.

	True	CLUS	KNN	LOF	SVM	RPCA	RCR
Blues	0	0/2	0/0	0/0	0/0	0/4	0/0
Classical	0	0/0	0/3	0/0	0/0	0/1	0/0
Country	4	2/0	2/0	3/0	0/0	2/0	1/2
Disco	7	5/0	5/0	2/0	0/0	4/0	5/2
Hip-hop	3	1/0	3/3	2/2	3/5	1/1	2/3
Jazz	2	0/7	1/6	1/5	0/0	0/5	2/0
Metal	17	1/0	2/0	5/1	14/0	2/2	2/1
Pop	4	4/10	2/2	2/11	2/8	3/3	2/0
Reggae	7	7/1	5/3	5/1	1/5	7/4	5/2
Rock	2	0/0	0/3	0/0	0/2	1/0	1/10

Table 3.5: Top 20 Ranked True Outliers/False Outliers Distribution Among Methods

3.7 Conclusion

In this work, we have demonstrated the application of outlier detection methods to a music dataset. Six state-of-the-art approaches to music genre recognition have been investigated, and their success is evaluated in terms of their ability to detect outliers found by human experts [19]. The results demonstrate that none of the methods can reliably identify outliers, in terms of accuracy. This leaves room for future improvement in the automatic detection of outliers in music data. Furthermore, the experiment results highlight the primary difficulties associated with outlier detection in music genre recognition. First, genre definitions are often subjective and ambiguous. Second, temporal dependencies of music need to be modeled. Third, low-level audio features can be a low entropy signal for capturing the high-level concepts. These challenges may also apply to other music datasets and should be further explored in future work.

We identify possible directions for future work as: First, as demonstrated in the experiments, improved feature representation should be beneficial for improved performance for the majority of the methods. Thus, enhancing feature representation for outlier detection algorithms suggests an important role for robustly isolating outliers. Second, given the temporal nature of music data, the static approach applied in the current framework might not be impractical. An outlier detection method that can handle the temporal dependencies could potentially boost performance. Third, the top 20 list for various methods reveals that distinct methods could be prone to the effects of different outlier types. For future research, an ensemble approach that incorporates several methods may be considered.

Chapter 4

Outlier Detection in Categorized Time-Series Datasets

4.1 Introduction

Outlier detection, one of the most frequently studied topics in data mining, is a task that aims to identify abnormal data points in a dataset of interest. Generally speaking, an outlier represents an instance that does not conform to the expected behavior and should, therefore, be subjected to further scrutiny. For example, in a security surveillance system [3], an outlier could be an intruder, whereas in credit card records [90], an outlier could be a fraudulent transaction. However, it is important to remember that although many real-world applications utilize outlier detection methods to discover unwanted behaviors, outliers are not always associated with negative impacts. This is why many studies refer to this technique as novelty detection, aiming to detect unusual data that could provide positive insights.

In recent decades, several methods have been proposed to detect outliers in various application domains, including financial services[90], transportation systems[70], and cyber-security[68].

Time-series data, unlike traditional data that often present as random samples of observations, represent data observed with a specific order. Time-series analysis is based on the assumption that successive observations in the data represent consecutive measurements, most commonly taken at equally spaced time intervals. Time-series classification (TSC) is a sub-domain that focuses on

classifying the various types of time series. Given the increased availability of time-series data, TSC problems have begun to attract considerable attention in recent years. Generally speaking, any classification problem that consists of a sequential observation can be formulated as a TSC problem. These applications of TSC now include human activity recognition [91], financial analyses [92], and music and video genre classification [93], among many others.

Outlier detection methods can also serve as a pre-processing tool to remove anomalous data. For example, Smith and Martinez[71] used a set of outlier detection methods to filter anomalies from a dataset in order to examine the effectiveness of outlier detection and reduction for improving the performance of several widely adopted classification methods. Their results indicated that removing outliers can lead to statistically significant improvements in both the training quality and the classification accuracy for most of the tested cases. These potential improvements are particularly appealing for the research community because not every dataset used in studies is noise-free. In fact, even datasets labeled by domain experts may contain errors. For example, the GTZAN dataset [7], a well-established dataset that has been utilized in several music genre classification studies, has been reported to contain erroneous genre labels [19]. Therefore, an outlier detection method that works on TSC data could potentially be applied to improve the integrity of a dataset and subsequently benefit downstream studies that utilize it.

Because of the inherent complexity of time series, developing a sophisticated approach to detect outliers in a time-series dataset is particularly a challenging research topic. Unlike many existing studies that focus on detecting abnormal activities within the time series, this paper aims instead to detect the abnormal associations between a time series and a subjective class label or an objective observation. This type of outlier detection problem must overcome the following challenges: 1) *Difficulties in modeling multi-dimensional time series*. The order within the sequence plays an important role when representing the abstract meaning of the data. Modeling the correlations among time-dependent observations introduces high complexity. 2) *Capturing the relationship between time series and class labels*. TSC data exhibits an input-output relationship between the time series and the class labels. Detecting the abnormal relationships in the data, therefore, requires that the patterns in the relationships be captured properly. 3) *Maintaining robustness against the large variations due to outliers*. Anomalous behavior can bias the model, preventing it from capturing the correct pattern if the modeling method does not take the robustness into account.

Existing outlier detection approaches focus on exploring the feature space, separating the instances,

and then identifying the instances that are isolated from the majority as outliers. However, with a complicated structure such as a multi-dimensional time series, optimizing the separation in an extremely large feature space becomes very difficult. In this paper, we propose a novel approach to expand the pattern from a seed to the entire dataset, enabling us to identify the outliers that deviate from the pattern of the majority of the dataset. The idea is to incorporate minimal human effort and offer an implicit definition of normality. The proposed framework can then learn the pattern and recognize normal behavior using this definition, so outliers that do not conform to the pattern can be identified. The contributions presented here include:

- **A novel semi-supervised framework**: A novel semi-supervised framework capable of performing general-purpose outlier detection on TSC data is proposed. Unlike the traditional outlier detection algorithms that often focus solely on the anomalous measures, our proposed framework aims to enhance the model's ability to recognize the normal behavior by expanding the pattern defined by a set of seed normal instances.
- **Integration of the CNN model to capture the abstract concept of time-series measurements**: The proposed approach addresses the two main challenges of detecting outliers in a TSC dataset, which are capturing the relationship between the time series and class labels and the temporal dependencies of the measurements in time series.
- **A robust outlier detection mechanism**: The proposed framework can tolerate outliers and remain unbiased when noise is present in the datasets. A single biased model will not significantly affect the overall expansion process by using a multi-arbitrator mechanism.
- **Extensive experiments to validate the effectiveness and efficiency of the framework**: Our experimental results demonstrate the effectiveness of our proposed approach for five real-world datasets from different domains. The advantages and limitations of the proposed approaches are also analyzed through a set of experiments.

The remainder of this paper is organized as follows. Section 4.2 reviews the existing work in the area of outlier detection on TSC data, and Section 4.3 presents the proposed framework for outlier detection. The experiments on real-world datasets are provided in Section 4.5. Finally, the paper concludes with a discussion of the findings in Section 4.7.

4.2 Related Work

Because existing studies on outlier detection in TSC datasets are relatively scarce, in this literature survey we instead focus on the two main components of this work, namely outlier detection and time series classification.

Outlier detection is an important subdomain for the data mining research community, and thousands of methods have been proposed to solve a wide variety of real-world problems. The existing outlier detection approaches can be categorized into five groups: distance-based [27], density-based [30], cluster-based [38], classification-based [28], and statistical-based [32] methods. Distance-based methods are one of the most common categories of outlier detection. An early distance-based study was the k-nearest neighbor (KNN) based method proposed by Ramaswamy et al. [27], which applies the distance criterion with KNN measures. Distance-based methods are usually both efficient and easy to implement, but their accuracy can be significantly affected when the data distribution is skewed. Derived from these early distance-based methods, density-based methods take the local density into account and thus outperform distance-based methods when it comes to detecting outliers in datasets that contain unevenly distributed groups in the feature space; local outlier factor (LOF) [30] is one of the most popular classic density-based methods. These methods can usually be applied easily to various types of datasets because of their density adaptive capability, but they do tend to suffer from higher computational overheads. Various outlier detection approaches, such as clustering and classification, have addressed this problem by transforming it into a traditional data mining problem. For example, Yu et al. proposed a clustering-based method [38] that first groups similar data and then labels the instances that cannot be adequately clustered as outliers. Some classification-based methods have also been proposed based on the assumption that the outlier identification process can be learned by a classification algorithm. For example, a one-class SVM-based method proposed by Das et al. [28] classifies each instance by incrementally learning a one-class SVM classifier that detects outliers utilizing a decision boundary. Statistical-based approaches are usually based on a data distribution assumption, detecting the outliers by identifying the instances assigned a low probability by the underlying assumption. These approaches often suffer from masking and swamping effects [94], however, as the outliers can bias the distribution parameters significantly during the estimation process, yielding biased probability distributions and causing normal instances to be misidentified as outliers and vice versa.

Given the broadness of outlier detection topics and the nature of most real-world applications that

involve the notion of order, a huge number of outlier detection techniques have been proposed for temporal datasets. Traditional research on time-series outlier detection aims to solve the following types of problems [95]: 1) detecting anomalous time series among a set of time series [96]; 2) detecting abnormal windows or points in the time series [97]; 3) detecting abnormal subsequences [98]. A variety of modeling and processing techniques have been proposed to capture patterns in temporal sequences, but the categorization of these approaches is very similar to the types of outlier detection methods described above.

Time-series classification focuses on the classification task for time-series datasets. With the explosive growth in time-correspondent applications in recent decades, many classification problems that involve sequential data that can be formulated as TSC tasks are widely studied. Many classification methods have been developed based on various underlying theories. In particular, one very popular group of methods involves ensemble approaches that incorporate the dynamic time wrapping (DTW) distance to capture the intra-timestamp correlations [99]. While some studies have utilized other approaches [100] to transform time-series data, most methods convert time series to a new representation and then apply a single or an ensemble group of classification methods accordingly. Based on this idea, Lines et al. proposed an ultimate ensembling method [101] that incorporates large number of classification methods and a variety of different feature transformation approaches. Another approach is to use deep neural network- (DNN) based methods. In recent years, deep learning has achieved significant success in machine learning-related domains such as computer vision [102] and natural language processing (NLP) [103]. In the TSC research community, a number of promising DNN-based solutions are also being proposed. Most of these apply convolutional neural networks (CNNs) [104], with a few exceptions using different architectures such as long-short term memory (LSTM) [105]; an empirical survey identified the promising performance of CNN for solving TSC problems [106], for example. Inspired by these studies, we chose to apply CNN as our main modeling method to capture the relationship between the time-series features and the class labels.

Specifically, the proposed method is a variant of conventional classification-based methods. Compared to existing outlier detection approaches, the proposed method aims to enhance the ability to identify normal instances by leveraging the seed normal instances while retaining the model's robustness against outliers using a multi-arbitrator mechanism.

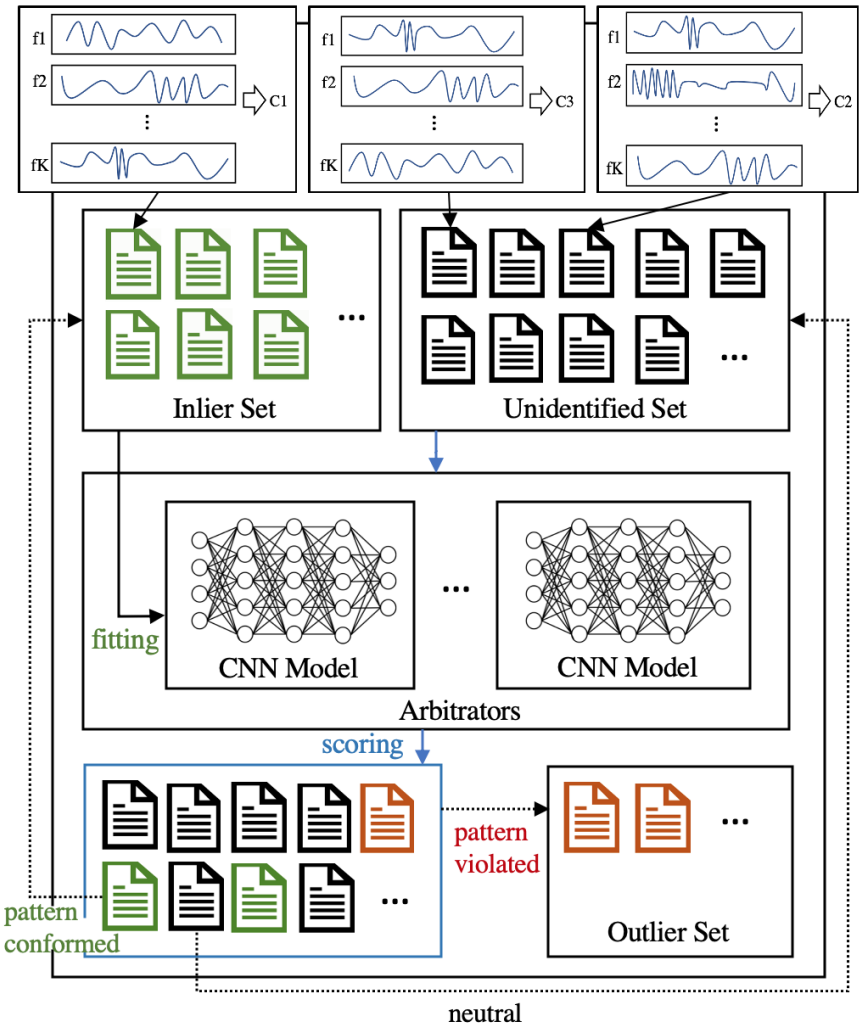


Figure 4.1: Dynamic Pattern Expansion Framework

4.3 Framework

The proposed framework consists of three main components, namely the CNN model, the arbitrators, and the DPE process. These three components are discussed in turn below.

4.3.1 CNN Model

CNN [107], one of the most effective architectures for deep learning, has been widely applied in computer vision, time series forecasting, and many other fields [107][104]. Generally speaking, CNN

can be used to model a variety of data, including time series, by capturing the inter-dependencies within the feature representations through a set of learnable filters. One advantage of CNNs over other popular modeling methods such as recurrent neural networks (RNNs), hidden Markov models (HMMs), and autoregressive (AR) models is that CNN models can take into account bidirectional relationships, whilst the time-series-focused methods can often only handle one specific direction. Another advantage of CNN over other state-of-the-art approaches is that CNNs have been shown to be more effective for feature extraction tasks that involve identifying abstract concepts in the datasets because of their primary assumption of strong local feature connectivity over the entire set of attributes of each instance. [108]

Architecture

We apply a deep CNN model to capture the TSC pattern. This architecture is similar to the model described in the work of Zhao et al. [104], which has delivered a promising performance for time-series classification. Our CNN model consists of an input layer, two pairs of convolutional layers and pooling layers, a fully connected layer, and the output layer. Figure 4.2 shows the architecture of this CNN model. The layers include:

1. Input transform layer: In this layer, we transform the time series data into the CNN representation. The dimension of this layer is $T \times K$, where T represents the length of the time series, and K denotes the number of features at each timestamp.

2. Convolutional layers: Each convolutional layer performs a convolution process by using a set of learnable filters to scan through the input to this layer. For time-series classification, 1-dimensional filters are applied to learn the relationship along the time axis of the feature representations. Several parameters must be predefined for this layer. The number-of-filters parameter F defines how many filters are initiated to learn the transformations; the stride s sets the length that will be moved for each step during the scan; the filter size W determines the window size of the filter when performing the scan. A larger W would allow the model to capture a longer temporal relationship of the time-series data.

3. Pooling layers: The purpose of each pooling layer is to downsample the output from the previous convolutional layer and thus reduce the number of model parameters. The operation

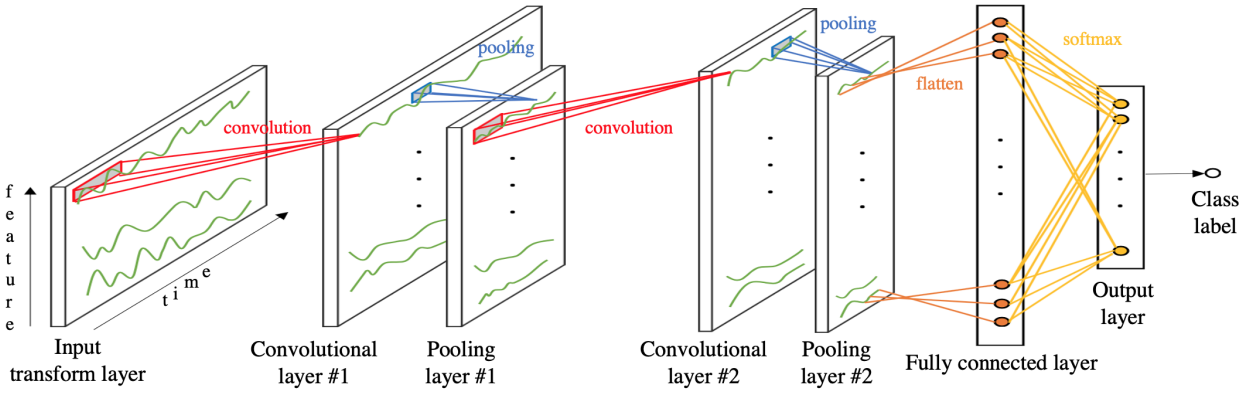


Figure 4.2: CNN Architecture

in this layer divides the input features to this layer into N segments, reducing each of the segments into a single value by taking the maximum.

4. Fully connected layer: This layer connects to all the activations in the previous layer, that is, the convoluted and downsampled model parameters. This is the layer that connects the learned representation from the time series with the class labels.

5. Output layer: This layer returns a softmax function that resembles the distribution of predicted classes.

4.3.2 Arbitrators

We introduce an arbitrator mechanism that detects outliers based on a majority vote from the arbitrators to maintain the robustness of the outlier detection task. Each arbitrator maintains a deep CNN model and gathers a different portion of the known normal instances. If an instance is identified as a normal instance by more than half of the arbitrators, it will be labeled as normal.

Each of the arbitrators takes a random sub-sample of known normal instances to train the CNN model. When the arbitrator is triggered, it trains the CNN model using the given portion of the normal instances and arbitrates the outlierness of each instance by performing a prediction on the remaining unknown instances. When an instance is voted as an outlier or a normal instance by the arbitrators, the outlierness score is defined by

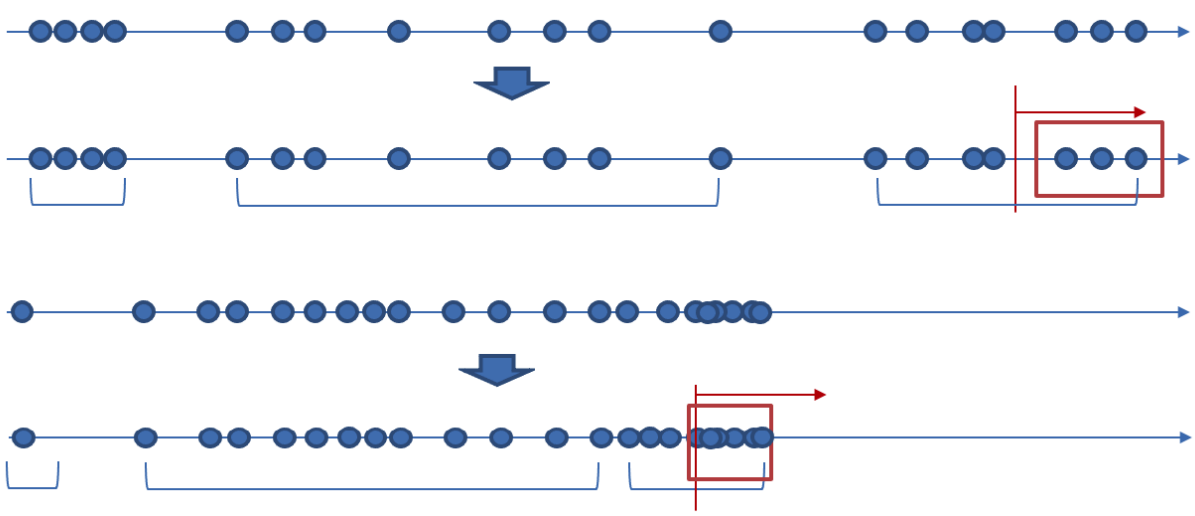


Figure 4.3: Adaptive Normal Identification Threshold

$$1 - softmax(z)_c \tag{4.1}$$

where z is the output of the fully connected layer, and c is the class label of the instance given by the input data. The softmax function is defined by

$$softmax(z)_c = \frac{e^{z_c}}{\sum_i^C e^{z_i}} \tag{4.2}$$

where C denotes the number of classes.

Each arbitrator labels normal instances with an adaptive threshold. As shown in Figure 4.3, the framework performs a Gaussian-mixture clustering on the softmax scores that the CNN model returned and adopts the mean value of the cluster that has the highest average score as the normal threshold. If the adaptive threshold is less than $1/C$, no normal instances will be labeled by this arbitrator. This mechanism ensures that the arbitrator to be effective in datasets that have different numbers of classes and avoids having a fixed normal instance threshold as a hyper-parameter.

4.3.3 Dynamic Pattern Expansion

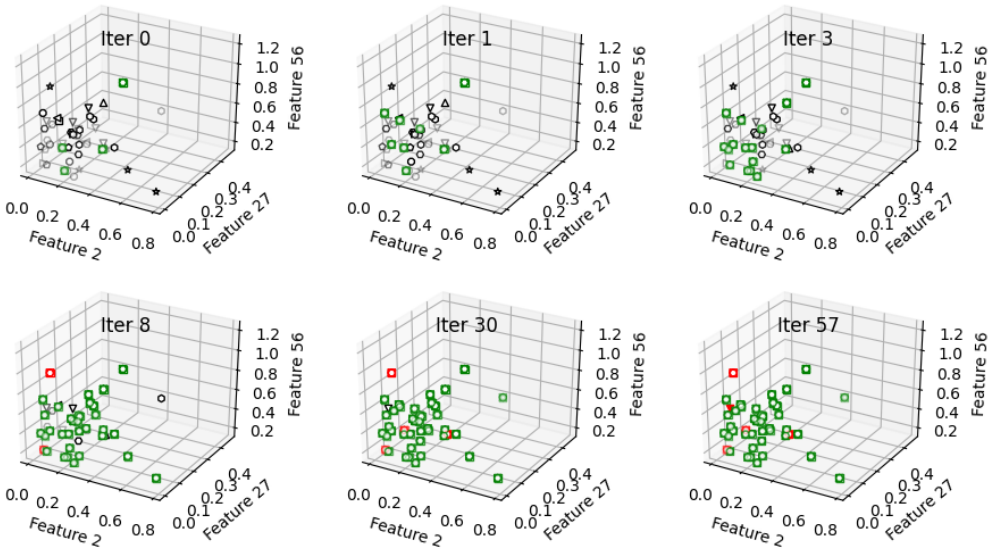


Figure 4.4: Pattern Expansion Over Iterations in the GTZAN Dataset

The dynamic pattern expansion (DPE) framework is presented as Algorithm 4. The framework starts with a set of inputs (lines 1–3), namely the feature matrix X, the class labels y, and the pre-selected seed normal instances S. With this initial set of real normal instances, the inner loop initiates an arbitrator with the CNN model to learn the pattern of the input-output relationship between the features and the class labels. The trained CNN models can then be used to predict the remaining unidentified samples (lines 8–10).

For each sample, if the softmax returns a probability that is greater than a certain threshold, the sample will be included in the true normal instance set in the next iteration; if the softmax function returns a value that is less than the exclusion threshold, the instance will be identified as an outlier and excluded from the unidentified candidate pool (lines 11–18). Here, the thresholds are heuristically selected based on the preknowledge. When the number of included samples exceeds the size limit for each arbitrator, the framework splits the samples by initiating a new arbitrator with the CNN model (lines 5–7). When there are more than two arbitrators, a majority voting method is applied to determine whether a sample should be included in the normal instance set or immediately labeled as an outlier (lines 11–18). If the voting results in a tie, the instance will remain in the unidentified set. The process will converge when no more normal instances are identified by the

Algorithm 4 Dynamic Pattern Expansion

Require: The class labels y, feature matrix X, and seeds S
Ensure: The anomalous instances
1: set $normalinstances = S$, $outliers = \{\}$
2: $R = allInstances \setminus S$
3: $converge = False$
4: **while not** $converge$ **do**
5: **for** 1 to $count(normal)/splitting - count(A)$ **do**
6: A.add(new Arbitrator)
7: **end for**
8: **for** a in A **do**
9: set $decision[a]$
10: $= a.arbitrate(subsample(normal), R, y, X)$
11: **end for**
12: **for** $instance$ in R **do**
13: **if** $majorityInclude(decision[a][instance])$ **for** a **in** A **then**
14: $normal$.add($instance$)
15: R.remove($instance$)
16: **else if** $majorityExclude(decision[a][instance])$ **for** a **in** A **then**
17: $outliers$.add($instance$)
18: R.remove($instance$)
19: **end if**
20: **end for**
21: **if** No new instance is added to $normal$ **then**
22: $converge = True$
23: **end if**
24: **end while**
25: **return** $outliers \cup R$

arbitrators in the inner loop (lines 20–22). Once the process reaches convergence, any remaining unidentified instances will be treated as outliers (line 24). After the detection process is completed, the framework fits the CNN model with all of the identified normal instances, including the seed normal instances, and uses this fitted model to perform a prediction on the outliers to generate the outlierness scores (Formula 4.1).

Figure 4.4 depicts the progress of this pattern expansion process. The green objects in the figure denote the instances that are identified as normal. In iteration 1, the only instances in the normal instance set are the seed normal instances; in subsequent iterations, the normal instance set grows while some of the other instances, shown in red, are identified as outliers. When there are no more normal instances to be identified, all the remaining instances are labeled as outliers.

Table 4.1: Average Detection Rate for Outlier Injected Data (Precision, Recall, F-Measure, AUC)

Method \ Data	GTZAN	HAR	Spoken Arabic Digits	Pen Digits	Phoneme
DPE-CNN	0.69, 0.88, 0.77, **0.99**	**0.98**, 0.47, 0.63, 0.97	0.39, 0.99, 0.56, **1.00**	0.41, 0.97, 0.57, **0.99**	0.08, **0.96**, 0.15, 0.64
DPE-AL	**0.71**, 0.88, **0.79**, **0.99**	0.55, 0.93, 0.69, **0.99**	**0.74**, **1.00**, **0.85**, **1.00**	0.45, 0.98, 0.62, **0.99**	**0.09**, 0.86, **0.16**, 0.63
DPE-GRU	0.24, **0.97**, 0.42, 0.98	0.27, **0.97**, 0.42, **0.99**	0.25, 0.95, 0.40, 0.98	0.38, **1.00**, 0.52, 0.95	0.08, 0.75, 0.14, **0.70**
LD	0.46, 0.48, 0.47, 0.90	0.63, 0.85, **0.73**, 0.97	0.55, 0.75, 0.63, 0.96	0.67, 0.94, **0.78**, **0.99**	0.05, 0.92, 0.10, 0.50
KNN	0.58, 0.57, 0.58, 0.95	0.63, 0.06, 0.11, 0.86	0.54, 0.06, 0.11, 0.97	**0.88**, 0.08, 0.15, **0.99**	0.08, 0.01, 0.02, 0.53
LOF	0.06, 0.05, 0.06, 0.51	0.06, 0.01, 0.01, 0.50	0.04, 0.01, 0.01, 0.55	0.05, 0.01, 0.01, 0.55	0.08, 0.01, 0.02, 0.50
SVM	0.05, 0.49, 0.09, 0.49	0.05, 0.10, 0.06, 0.50	0.05, 0.34, 0.09, 0.51	0.05, 0.61, 0.09, 0.51	0.04, 0.35, 0.07, 0.47

4.4 Active Learning Framework

We further extended the DPE framework with an active learning strategy. In each iteration, the model identifies the most valuable unknown instances and queries the oracle. The oracle answers with the outlier labels, and the framework can either put them into the normal instances set or outliers set, depending on the answer from the oracle. In the next round, the model can be improved by using the ground truth labels in the fitting process.

The importance ranking strategy is composed by two ideas: 1) The most unsure instances, and 2) The instances that are similar to most of the other instances. The overall query strategy is defined by the importance measure

$$m_i = \frac{var(scores_i)}{median(dist_i)} \tag{4.3}$$

where each $scores_i$ is the score of the arbitrators' softmax prediction score on instance i, and $dist_i$ represents the Euclidean distance from instance i to its k-nearest neighbors.

4.5 Experiment

The experiment was conducted on five sets of real-world, time-series classification data; for each dataset, the performance of our proposed method was compared to that achieved by five benchmark methods. The experiment was run on a Ubuntu machine with Intel i7 1.7GHz CPU, 32GB RAM, and an Nvidia GTX 1080ti graphics card.

4.5.1 Benchmark Methods

To the best of our knowledge, no outlier detection method is specifically designed for TSC datasets. In addition to one benchmark method that also utilizes the DPE framework, we selected one classification-dataset-specific outlier detection system and three additional non-classification-dataset-specific outlier detection systems as benchmark methods. Since none of these methods were designed for time-series data, we aggregated the time series by computing the mean and variance across the time axis before applying these methods. The methods used were:

1. **DPE framework with gated recurrent unit (DPE-GRU)**, a benchmark method that combines the DPE framework with a gated recurrent unit (GRU) [109]. We set up a baseline model to compare the performance with our CNN model. GRU is a commonly used RNN structure similar to the long-short term memory (LSTM). It has been experimentally shown to achieve comparable performance as LSTM with improved efficiency given its less complex structure. Here, we also utilized global max pooling, which is a widely applied pooling method for temporal tasks.

2. **Lineardetection (LD)** [66], a robust logistic-regression-based outlier detection method that captures the input-output relationship between the features and the class labels. The detection process fits a logistic regression with a heavy-tailed error assumption with variational Bayesian inference.

3. **KNN method** [83], a distance-based outlier detection method that estimates the level of outlierness by calculating the distance of each instance to the k nearest neighbors.

4. **Local outlier factor (LOF) method** [74], an extension of the KNN method that takes the local densities of each instance into account. For each data instance, the local outlier factor is estimated based on the density close to the object, and the outlierness score can thus be estimated.

5. **One-class SVM method** [84], first developed as an unsupervised classification method that aims to identify whether the new incoming data instances are in-class or out-class. This method is often used as an outlier detection method when the first instance the model acquires is a normal instance.

The evaluation was based on four standard metrics: precision, recall, F-measure, and area under the ROC curve (AUC).

4.5.2 Benchmark Datasets

We conducted experiments in five real-world datasets to verify the performance of DPE:

1. GTZAN dataset [7]: GTZAN is one of the most frequently studied datasets for the MIR research community, serving as a classic benchmark dataset for the MGR task. This dataset consists of 10 genres. Each genre includes 100 files for analysis, and each file consists of a 30-second long excerpt. In this dataset, Sturm identified a set of mislabeled and corrupted music clips in a previous study [19]. These problematic labels, which are undesirable for training and measuring the performance of MGR systems, came in useful for this experiment as they allow us to apply our proposed approach to detect the mislabeled and corrupted music clips. Because the dataset only contains a set of music clips and class labels, a pre-processing step was required to convert the music clips into numerical features. We applied AudioSet, a pre-trained CNN-based model [110] to extract features.

2. Human activity recognition (HAR): The HAR dataset is a record of the activities of 30 volunteers, monitored via a smartphone worn on the waist. The class labels identify six different types of activities: walking, walking upstairs, walking downstairs, sitting, standing, and lying down. The dataset consists of 10299 time series; each has 128 timestamps and nine features on each timestamp.

3. Spoken Arabic digits [111]: This dataset contains extracted features of mel-frequency cepstrum coefficients (MFCCs) corresponding to spoken Arabic digits from 44 male and 44 female Arabic speakers. There are 8800 instances, and each instance consists of 93 timestamps and 13 features on each timestamp.

4. Pen digits [112]: The pen digit dataset contains 10992 instances collected from 44 writers' handwriting data, with each instance represents a 2-dimensional trajectory (x_t and y_t on the plane) that corresponds to a digit. It has only eight timestamps, which is the shortest sequence length in our benchmark datasets.

5. Phoneme [113]: The Phoneme dataset consists of 6668 instances of processed audio features of phonetic time series that were collected from Google translate. The dataset has 39 classes, and each instance is a 217-timestamp-long multivariate representation that represents 11 features extracted from various frequency bands of the spectrum.

4.5.3 Experimental Results

Two sets of experiments were conducted. In each of the DPE runs, 10 true normal instances from each class have been randomly chosen as the seed normal instance set. In the first set of experiments, we conducted the experiments on all five benchmark datasets. We applied the approach described in [32] and [34] to artificially inject outliers; a contamination process is performed by randomly choosing 5% of instances and randomizing their class labels to create outliers for each chosen instance. For the GTZAN dataset, which already contains expert-identified outliers, we removed those known outliers and then injected artificial outliers with the same contamination process. The performance metrics were calculated by the average of 10 sets of different randomly contaminated datasets. As shown in Table 4.1, the proposed method outperformed the other benchmark methods in most of the tested datasets on most of the evaluation metrics. DPE-CNN and DPE-GRU showed advantages on the datasets that have a larger number of class labels and longer sequence length. This result demonstrates the ability of DPE to process large temporal datasets and capture complex patterns while maintaining robustness. Here, DPE-CNN did not present a significant advantage over DPE-GRU in terms of the performance metrics. However, in the experiments on datasets with longer sequence length, DPE-GRU suffered from a much longer model fitting time because its RNN architecture requires more sequential processes than the CNN-based approach.

We also compiled a confusion matrix for the injected outliers that *DPE* was not able to detect in the 10 sets of outlier-injected GTZAN tests. As shown in Figure 4.7, the observations match our knowledge regarding music genre classification. There are certain levels of ambiguity when it comes to differentiating between *Rock*, *Blues* and *Country* music as well as between *Blues* versus *Reggae*. *Blues* and *Jazz* share many common characteristics, as do *Raggae*, *Disco*, and *Hip-hop*.

In the second set of experiments, we tested the ability of DPE and the other benchmark methods to detect the real outliers identified by Sturm [19] in the GTZAN dataset. The results are shown in Table 4.2. Although the overall performance was not as strong as in the injected experiment, DPE

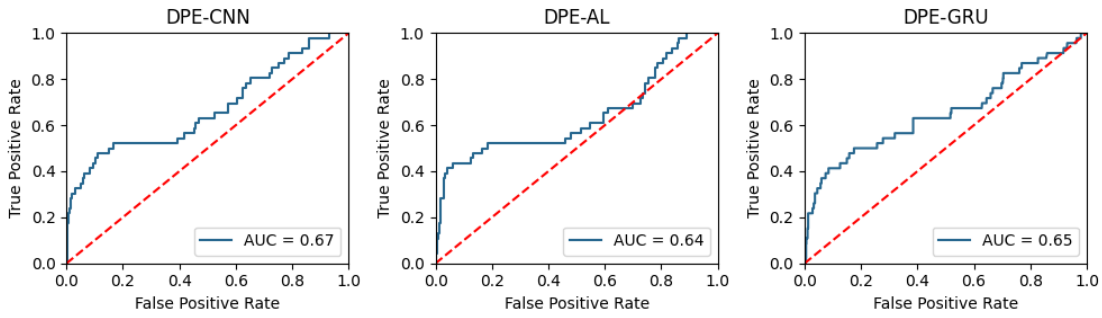


Figure 4.5: Selected ROC Curves for Detecting GTZAN Expert-Identified Outliers of DPE Methods

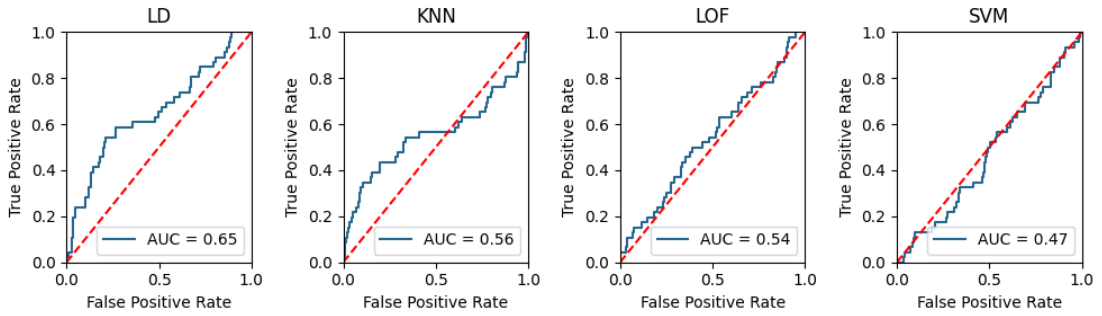


Figure 4.6: Selected ROC Curves for Detecting GTZAN Expert-Identified Outliers of Benchmark Methods

still significantly outperformed the benchmark methods in all metrics. This outcome is different from the results of the first experiment with artificially injected outliers, probably because the expert-identified outliers do not present dissimilarities as clear as the outliers generated from random swapping in the feature space. Two clips from different genres may actually share the same characteristics because music genre is a subjective classification. As shown in Figure 4.7, it is more difficult to discriminate between the genres that have common characteristics. Furthermore, it is also possible that the perceptual difference between the normal instances and the expert-identified outliers are not captured by the AudioSet-based features and thus result in the sub-optimal performance. Figure 4.5 and 4.6 shows the ROC curves. Although DPE achieved higher precision and recall, the ROC curve did not show any significant advantages compared to the leading methods. The ROC curves of the DPE methods tend to increase TPR to around 0.5 when the FPR is low and then become flat in the 0.25 to 0.5 FPR region before once again increasing slowly toward the upper right corner above 0.5 FPR. This type of curve suggests the outstanding effectiveness of the method when the outlierness score returned is high. However, when the returned outlierness score is close to the

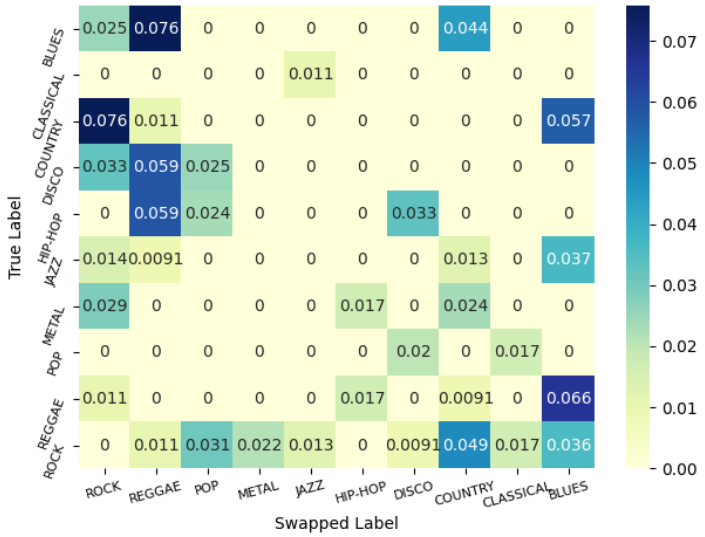


Figure 4.7: False Negative Confusion Matrix in GTZAN Injected Experiment

Table 4.2: Performance Comparison on Real GTZAN Abnormal Clip Detection

	Precision	Recall	F-Measure	AUC
DPE-CNN	**0.34**	0.30	0.32	**0.67**
DPE-AL	0.25	**0.48**	**0.33**	0.64
DPE-GRU	0.16	0.46	0.23	0.65
LD	0.20	0.20	0.20	0.65
KNN	0.16	0.17	0.17	0.56
LOF	0.10	0.11	0.10	0.54
SVM	0.04	0.35	0.07	0.47

overall median, many real normal instances are incorrectly scored as being higher than they should be. Based on our observations, this scenario occurs when DPE mistakenly includes a real outlier into the normal set, which leads to the wrong pattern being expanded in subsequent iterations.

4.6 Hyper-parameter Impact Analysis

We conducted a set of experiments that varies on only one hyper-parameter and observe the performances on each dataset to analyze the impact of the hyper-parameters of DPE-CNN. The results are an average from 10 test datasets. Each experiment is conducted varying only one hyper-parameter and fixing all others. Figure 4.8 shows the comparison results. The number of seed normal instances show an obvious impact on the performance. However, the framework

achieved acceptable performance when starting with 10 seed normal instances for each class. The outlier identification threshold is another hyper-parameter that strongly affects performance. The experimental results suggest setting this threshold to a value less than 0.001. The learning rate of the CNN model presents the best performance when it is set to 0.0005 to 0.001, and the F-measure significantly dropped when the learning rate is set to 0.0001. The best results were presented when the number of arbitrators is set to 2. Interestingly, when there is only a single arbitrator, the performance was not significantly affected. In contrast, increasing the number of arbitrators does not show performance improvements. The model training epoch and dropout rate did not show obvious impacts when they were varied. A dropout rate of 0.3 produced the best results in our experiment. A number of epochs equal to 50 was sufficient in the test datasets, and fitting with more epochs did not show significant improvements to the F-measure.

4.7 Conclusion

In this work, we presented a novel CNN-based outlier detection framework that requires only a small number of seed normal samples. The framework expands the pattern throughout the entire dataset, identifying all the normal instances and outliers iteratively. The proposed method demonstrated promising results and outperformed five benchmark methods on five real-world, time-series datasets from different domains. During the pattern expansion process, the framework maintained its robustness against outliers. The proposed method was able to detect most of the outliers in the injected experiment datasets using a few carefully validated seed normal instances.

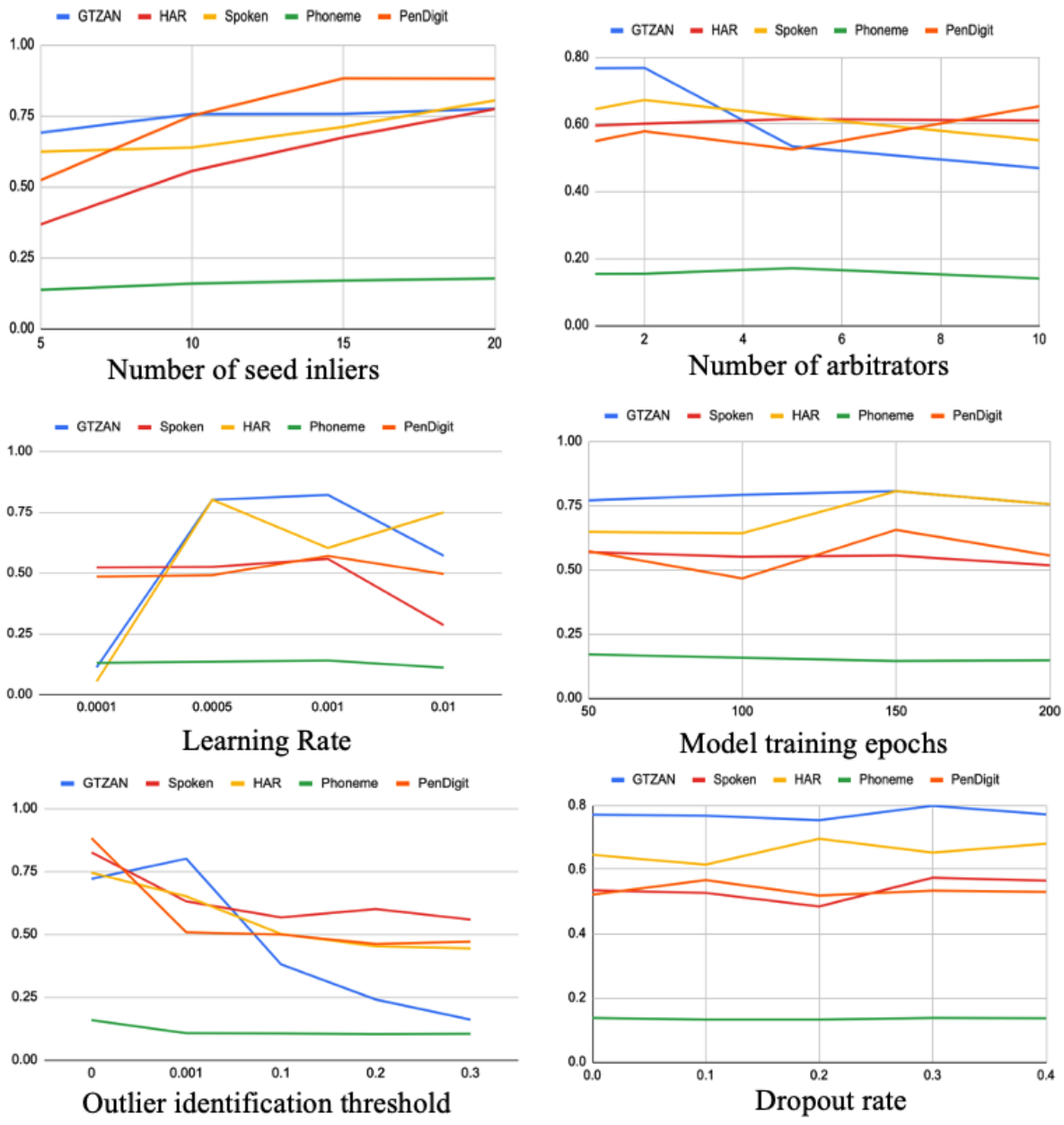


Figure 4.8: Framework Hyper-parameter Impact Comparison (F-measure)

Chapter 5

Completed Work and Future Work

5.1 Research Tasks

The major research tasks are described as follows. The current status of these tasks is listed in Table 5.1.

5.1.1 Outlier Detection in Mixed-type Datasets

- **Development of an unsupervised framework for mixed-type outlier detection (A1)** We propose a novel unsupervised approach for general outlier detection in mixed-type data. Our method requires no intensive human labor, such as training set labeling.
- **Design of a novel robust model for mixed-type data (A2)** We design a novel regression-based model for capturing the pattern among the mixed-type data attributes, incorporating the techniques of the generalized linear model (GLM) and robust estimation.
- **Derivation of approximate inference method (A3)** We propose an approach to approximate the intractable Bayesian inference for the proposed model.
- **Experimental justification of the proposed framework (A4)** A set of experiments is conducted in synthetic datasets and various public, real-world datasets demonstrating the significant improvement in accuracy from the benchmark methods.

- **Extension of a new approximate inference method (A5)** An alternative Bayesian inference approach is developed, aiming to provide improved efficiency and better approximation accuracy.
- **Extension of experiments (A6)** A more extensive set of experiments is conducted on three new large real data sets to demonstrate the efficiency and effectiveness of our framework. New synthetic experiments are also conducted, including time cost analysis and parameter impact analysis to show the capability of the model.

5.1.2 Categorical Outlier Detection in Music Genre Datasets

- **Design of an unsupervised outlier detection framework for music genre datasets (B1)** We raise this novel application to the music information retrieval community and design an unsupervised framework to address the underlying issue. The framework includes feature extraction that turns the music clips into numerical features and anomaly detection that identifies the distorted or misclassified music clips.
- **Proposal of categorical outlier detection model (B2)** A robust categorical regression-based model is proposed to improve the outlier detection accuracy for this specific application.
- **Development of parameter inference algorithm (B3)** Because the model is analytically intractable, we develop a variational inference-based approach to approximate the Bayesian inference.
- **Extensions of new parameter inference algorithm (B4)** Because the model is not set by conjugate prior, the variational inference-based approach is losing accuracy on more approximation with IRLS. Therefore, an MCMC-based approach is proposed to address this issue.

5.1.3 Detecting Outliers in Categorized Time-Series Data

- **Design of new outlier detection framework for categorized time-series data (C1)** Design a novel, semi-supervised framework for outlier detection on categorized time-series data.

Table 5.1: Research Tasks and Status

Task	Description	Status
Research Area A	Outlier Detection in Mixed-type Datasets	Completed
A1	Development of an unsupervised framework for mixed-type outlier detection	
A2	Design of a novel robust model for mixed-type data	
A3	Derivation of approximate inference method	
A4	Experimental justification of the proposed framework	
A5	Extension of new approximate inference method	
A6	Extension of experiments	
Research Area B	Categorical Outlier Detection in Music Genre Datasets	Completed
B1	Design of an unsupervised outlier detection framework for music genre datasets	Completed
B2	Proposal of categorical outlier detection methods	Completed
B3	Proposal of parameter inference algorithm	Completed
B4	Extension to new parameter inference algorithm	Completed
B5	Extensions of the experiments	Completed
Research Area C	Detecting Outliers in Categorized Time Series Data	Completed
C1	Design of new outlier detection framework for categorized time series data	Completed
C2	Proposal of modeling strategy for categorized time-series data	Completed
C3	Validate performance on real-world datasets	Completed
C4	Extension of the modeling method	Completed
C5	Extension of the experiments on more real-world datasets	Completed
C6	Extension of the experiments of hyper-parameter impact analysis	Completed
Thesis Revision D	Thesis Revision	Completed

- **Proposal of modeling strategy for categorized time-series data (C2)** Proposed a modeling method that captures the sequential information of the time series and the relationship between the time series and the category labels

- **Validate performance on real-world datasets (C3)** Design a set of experiments to analyze the performance of the proposed framework. Conduct the experiments in real datasets to prove the effectiveness of the approach.

- **Extension of the modeling method (C4)** Extend the modeling method to explore the possibilities of applying other based classifiers in the framework.

- **Extension of the experiments on more real-world datasets (C5)** Extend the experiments to run on more real-world, categorized time-series data and analyze the experiment results.

- **Extension of the experiments of parameter impact analysis (C6)** Conduct a comprehensive parameter impact analysis to the framework and provide a guideline for finding the optimal set of parameter settings on various types of scenarios.

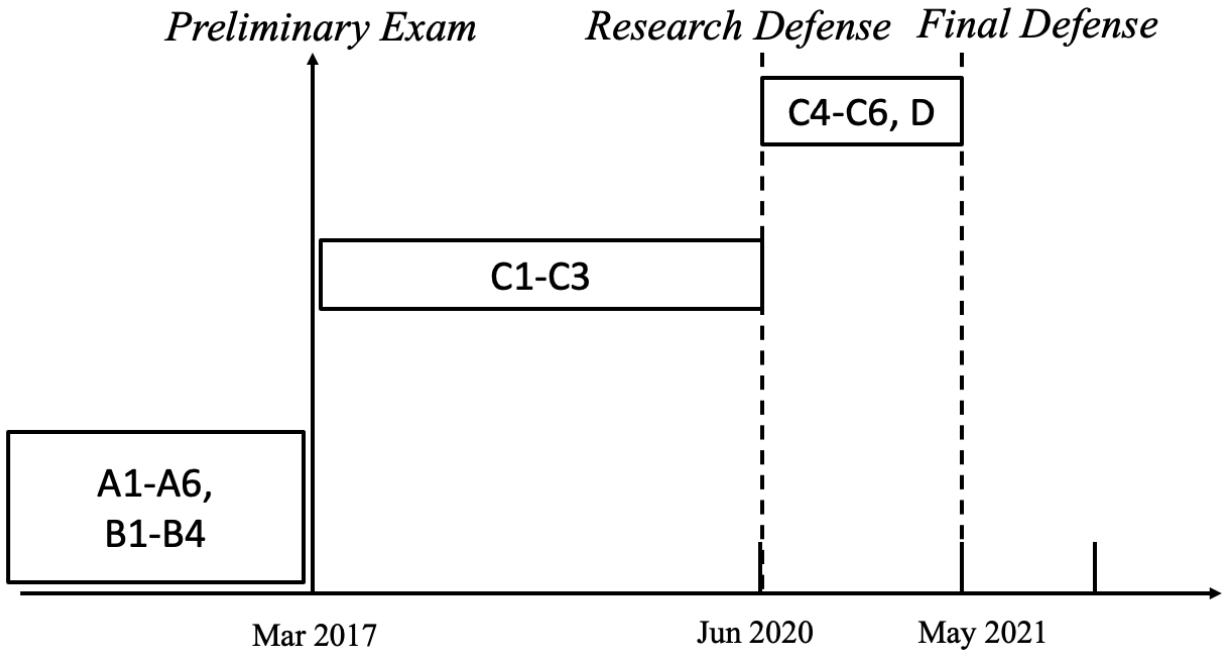


Figure 5.1: Research Tasks Schedule

5.2 Schedule

The proposed work includes 18 specific tasks, including thesis revision. All the tasks are finished by May 2021. The schedule of the proposed research is illustrated in Figure 5.1

5.3 Current Publications

5.3.1 Published Papers

Journal Papers

1. Yen-Cheng Lu, Feng Chen, Yating Wang, Chang-Tien Lu, "Discovering Anomalies on Mixed-Type Data using a Generalized Student-t Based Approach," IEEE Transactions on Knowledge and Data Engineering (TKDE), vol. 28 no. 10, pp. 2582–2595, 2016

2. Yating Wang, Ing-Ray Chen, Jin-Hee Cho, Ananthram Swami, Yen-Cheng Lu, Chang-Tien

Lu, Jeffery Tasi, "CATrust: Context-Aware Trust Management for Service-Oriented Ad Hoc Networks," IEEE Transactions on Services Computing (TSC), vol.PP, no.99, pp.1–1, 2016

3. Xutong Liu, Feng Chen, Yen-Cheng Lu, and Chang-Tien Lu. 2017. "Spatial Prediction for Multivariate Non-Gaussian Data," ACM Transactions on Knowledge Discovery Data 11, 3, Article 36, April 2017

Conference Papers

1. Yen-Cheng Lu, Feng Chen, Yang Chen and Chang-Tien Lu, "A Generalized Student-t Based Approach to Mixed-Type Anomaly Detection," in Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence (AAAI 2013), pp. 633–639, full paper (acceptance rate: 29%), Bellevue, Washington, July 2013

2. Yen-Cheng Lu, Chih-Wei Wu, Chang-Tien Lu and Alexander Lerch, "An Unsupervised Approach to Anomaly Detection in Music Datasets," in Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2016), pp. 749–752, short paper (acceptance rate: 30%), Pisa, Italy, July 2016

3. Yen-Cheng Lu, Chih-Wei Wu, Chang-Tien Lu and Alexander Lerch, "Automatic Outlier Detection in Music Genre Datasets," in Proceedings of the 17th International Society for Music Information Retrieval Conference (ISMIR 2016), pp. 101–107, full poster paper, New York, August 2016

4. Kaiqun Fu, Yen-Cheng Lu, Chang-Tien Lu, "TREADS: A Safe Route Recommender using Social Media Mining and Text Summarization," in Proceedings of the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (ACMGIS 2014), pp. 557–560, demo paper, Dallas, Texas, November 2014

5. Yating Wang, Yen-Cheng Lu, Ing-Ray Chen, Jin-Hee Cho, Ananthram Swami and Chang-Tien Lu, "LogitTrust: A Logit Regression-based Trust Model for Mobile Ad Hoc Networks," in Proceedings of the Sixth ASE International Conference on Privacy, Security, Risk and Trust (PASSAT 2014), full paper, Cambridge, Massachusetts, December 2014

6. Zhiqian Chen, Chih-Wei Wu, Yen-Cheng Lu, Chang-Tien Lu and Alexander Lerch, "Learning to Fuse Music Genres with Generative Adversarial Dual Learning," 17th IEEE International Conference on Data Mining (ICDM 2017), pp. 81–822, New Orleans, USA, November 2017

5.3.2 Submitted and In-preparation Papers

1. Yen-Cheng Lu, Yao-Chun Chan, and Chang-Tien Lu, “A CNN-based Pattern Expansion Approach for Outlier Detection,” 37th Conference on Uncertainty in Artificial Intelligence (UAI 2021) (Submitted)

2. Yen-Cheng Lu and Chang-Tien Lu, ”Active Dynamic Pattern Expansion for Relational Anomaly Detection in Temporal Datasets,” 30th ACM International Conference on Information and Knowledge Management (CIKM 2021) (In preparation)

Bibliography

[1] Clay Spence, Lucas Parra, and Paul Sajda. Detection, synthesis and compression in mammographic image analysis with a hierarchical image probability model. In *Proceedings of the IEEE Workshop on Mathematical Methods in Biomedical Image Analysis (MMBIA'01)*, pages 3–10, Washington, DC, USA, 2001. IEEE Computer Society.

[2] V. Kumar. Parallel and distributed computing for cybersecurity. *IEEE Distributed Systems Online*, 6(10):1, 2005.

[3] T. Brotherton and T. Johnson. Anomaly detection for advanced military aircraft using neural networks. In *Aerospace Conference, 2001, IEEE Proceedings.*, volume 6, pages 3113–3123, 2001.

[4] E. Aleskerov, B. Freisleben, and B. Rao. Cardwatch: a neural network based database mining system for credit card fraud detection. In *Computational Intelligence for Financial Engineering (CIFEr), 1997., Proceedings of the IEEE/IAFE 1997*, pages 220–226, Mar 1997.

[5] Peter Szor. *The Art of Computer Virus Research and Defense*. Addison-Wesley Professional, 2005.

[6] Stephan Fischer, Rainer Lienhart, and Wolfgang Effelsberg. Automatic recognition of film genres. In *Proceedings of the Third ACM International Conference on Multimedia*, MULTIMEDIA '95, pages 295–304, New York, NY, USA, 1995. ACM.

[7] George Tzanetakis and Perry Cook. Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing*, 10(5):293–302, 2002.

[8] Marina Ivasic-Kos, Miran Pobar, and Ivo Ipsic. *Automatic Movie Posters Classification into Genres*, pages 319–328. Springer International Publishing, Cham, 2015.

[9] V. M. T. Ochoa, S. Y. Yayilgan, and F. A. Cheikh. Adult video content detection using machine learning techniques. In *2012 Eighth International Conference on Signal Image Technology and Internet Based Systems*, pages 967–974, Nov 2012.

[10] Christian Jansohn, Adrian Ulges, and Thomas M. Breuel. Detecting pornographic video content by combining image features with motion information. In *Proceedings of the 17th ACM International Conference on Multimedia*, MM '09, pages 601–604, New York, NY, USA, 2009. ACM.

[11] C. Y. Kim, O. J. Kwon, W. G. Kim, and S. R. Choi. Automatic system for filtering obscene video. In *2008 10th International Conference on Advanced Communication Technology*, volume 2, pages 1435–1438, Feb 2008.

[12] Rehanullah Khan, Julian Stöttinger, and Martin Kampel. An adaptive multiple model approach for fast content-based skin detection in on-line videos. In *Proceedings of the 1st ACM Workshop on Analysis and Retrieval of Events/Actions and Workflows in Video Streams*, AREA '08, pages 89–96, New York, NY, USA, 2008. ACM.

[13] O. Deniz, I. Serrano, G. Bueno, and T. K. Kim. Fast violence detection in video. In *2014 International Conference on Computer Vision Theory and Applications (VISAPP)*, volume 2, pages 478–485, Jan 2014.

[14] Richard J. Bolton and David J. Hand. Statistical fraud detection: A review. *Statistical Science*, 17(3):235–249, 2002.

[15] Tom Fawcett and Foster Provost. Adaptive fraud detection. *Data Mining and Knowledge Discovery*, 1(3):291–316, 1997.

[16] X. Wang and G. Dong. Research on money laundering detection based on improved minimum spanning tree clustering and its application. In *2009 Second International Symposium on Knowledge Acquisition and Modeling*, volume 2, pages 62–64, Nov 2009.

[17] Amol Ghoting, Matthew Eric Otey, Srinivasan Parthasarathy, and The Ohio. Loaded: Link-based outlier and anomaly detection in evolving data sets. In *Proceedings of the 4th IEEE International Conference of Data Mining*, pages 387–390, 2004.

[18] M. E. Otey, S. Parthasarathy, and A. Ghoting. Fast lightweight outlier detection in mixed-attribute data sets. *DMKD*, 2006.

[19] Bob L. Sturm. An analysis of the GTZAN music genre dataset. In *Proceedings of the second international ACM workshop on Music Information Retrieval with user-centered and multimodal strategies (MIRUM)*, 2012.

[20] M. Shinozuka George Deodatis. Auto‐regressive model for nonstationary stochastic processes. *Journal of Engineering Mechanics*, 114(11):1995–2012, Issue: object: doi:10.1061/jenmdt.1988.114.issue-11, revision: rev:1479266996475-21598:doi:10.1061/jenmdt.1988.114.issue-11, .

[21] S. P. Chatzis, D. I. Kosmopoulos, and T. A. Varvarigou. Robust sequential data modeling using an outlier tolerant hidden markov model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(9):1657–1669, Sept 2009.

[22] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, November 1997.

[23] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhudinov. Unsupervised learning of video representations using lstms. In David Blei and Francis Bach, editors, *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pages 843–852. JMLR Workshop and Conference Proceedings, 2015.

[24] Yen-Cheng Lu, Feng Chen, Yang Chen, and Chang-Tien Lu. A generalized student-t based approach to mixed-type anomaly detection. *Proceedings of the AAAI Conference on Artificial Intelligence*, 27(1), Jun. 2013.

[25] Yen-Cheng Lu, Feng Chen, Yating Wang, and Chang-Tien Lu. Discovering anomalies on mixed-type data using a generalized student- t based approach. *IEEE Transactions on Knowledge and Data Engineering*, 28(10):2582–2595, 2016.

[26] Edwin M. Knorr, Raymond T. Ng, and Vladimir Tucakov. Distance-based outliers: algorithms and applications. *The VLDB Journal*, 8(3–4):237–253, 2000.

[27] Sridhar Ramaswamy, Rajeev Rastogi, and Kyuseok Shim. Efficient algorithms for mining outliers from large data sets. *SIGMOD Record*, 29(2):427–438, May 2000.

[28] Santanu Das, Bryan L. Matthews, Ashok N. Srivastava, and Nikunj C. Oza. Multiple kernel learning for heterogeneous anomaly detection: algorithm and aviation safety case study. In *KDD 10': 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 47–56, 2010.

[29] Volker Roth. Outlier detection with one-class kernel fisher discriminants. In *Advances in Neural Information Processing Systems 17*, pages 1169–1176, 2005.

[30] Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, and Jörg Sander. Lof: identifying density-based local outliers. *SIGMOD Record*, 29(2):93–104, May 2000.

[31] S. Papadimitriou, H. Kitagawa, P.B. Gibbons, and C. Faloutsos. Loci: fast outlier detection using the local correlation integral. In *Data Engineering, 2003. Proceedings. 19th International Conference on*, pages 315–326, March 2003.

[32] Marco Riani, Anthony C. Atkinson, and Andrea Cerioli. Finding an unknown number of multivariate outliers. *Journal of the Royal Stats Society Series B*, 71(2):447–466, 2009.

[33] Shohei Hido, Yuta Tsuboi, Hisashi Kashima, Masashi Sugiyama, and Takafumi Kanamori. Statistical outlier detection using direct density ratio estimation. *Knowledge Information Systems*, 2011.

[34] Andrea Cerioli. Multivariate outlier detection with high-breakdown estimators. *Journal of the American Statistical Association*, 105(489):147–156, 2009.

[35] Gregory Piatetsky-Shapiro, Chabane Djeraba, Lise Getoor, Robert Grossman, Ronen Feldman, and Mohammed Zaki. What are the grand challenges for data mining?: Kdd-2006 panel report. *SIGKDD Explor. Newsl.*, 8(2):70–77, 2006.

[36] Qiang Yang and Xindong Wu. 10 challenging problems in data mining research. *International Journal of Information Technology and Decision Making*, 5(4):597–604, 2006.

[37] Thomas P. Minka. Expectation propagation for approximate Bayesian inference. In *Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence*, UAI '01, pages 362–369, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.

[38] Dantong Yu, Gholam Sheikholeslami, and Aidong Zhang. Findout: Finding outliers in very large datasets. Technical report, Dept. of CSE SUNY Buffalo, 1999.

[39] David E. Tyler. Robust statistics: Theory and methods. *Journal of the American Statistical Association*, 103:888–889, 2008.

[40] Mei-Ling Shyu, Shu-Ching Chen, Kanoksri Sarinnapakorn, and LiWu Chang. A novel anomaly detection scheme based on principal component classifier. Technical report, DTIC Document, 2003.

[41] Cláudia Pascoal, M Rosario de Oliveira, Rui Valadas, Peter Filzmoser, Paulo Salvador, and António Pacheco. Robust feature selection and robust pca for internet traffic anomaly detection. In *INFOCOM, 2012 Proceedings IEEE*, pages 1755–1763. IEEE, 2012.

[42] Emmanuel J Candès, Xiaodong Li, Yi Ma, and John Wright. Robust principal component analysis? *Journal of the ACM (JACM)*, 58(3):11, 2011.

[43] Ling Huang, XuanLong Nguyen, Minos Garofalakis, Michael I Jordan, Anthony Joseph, and Nina Taft. In-network pca and anomaly detection. In *Advances in Neural Information Processing Systems*, pages 617–624, 2006.

[44] Dorian Pyle. *Data preparation for data mining*, volume 1. Morgan Kaufmann, 1999.

[45] Truyen Tran, Dinh Phung, and Svetha Venkatesh. Mixed-variate restricted boltzmann machines. In *Proceedings of 3rd Asian Conference on Machine Learning (ACML)*, 2011.

[46] Anna Koufakou, Michael Georgiopoulos, and Georgios C Anagnostopoulos. Detecting outliers in high-dimensional datasets with mixed attributes. In *DMIN*, pages 427–433, 2008.

[47] Khoi-Nguyen Tran and Huidong Jin. Detecting network anomalies in mixed-attribute data sets. In *IEEE Third International Conference on Knowledge Discovery and Data Mining (WKDD'10).*, pages 383–386, 2010.

[48] Anna Koufakou and Michael Georgiopoulos. A fast outlier detection strategy for distributed high-dimensional data sets with mixed attributes. *Data Mining and Knowledge Discovery*, 20(2):259–289, 2010.

[49] Mao Ye, Xue Li, and Maria E Orlowska. Projected outlier detection in high-dimensional mixed-attributes data set. *Expert Systems with Applications*, 36(3):7104–7113, 2009.

[50] Ke Zhang and Huidong Jin. An effective pattern based outlier detection approach for mixed attribute data. In *AI 2010: Advances in Artificial Intelligence*, pages 122–131. Springer, 2011.

[51] Brad Warner and Manavendra Misra. Understanding neural networks as statistical tools. *The American Statistician*, 50(4):284–293.

[52] P. J. Rousseeuw and A. M. Leroy. *Robust Regression and Outlier Detection*. John Wiley & Sons, Inc., New York, NY, USA, 1987.

[53] Andrew Gelman, John B. Carlin, Hal S. Stern, and Donald B. Rubin. *Bayesian Data Analysis*. Chapman and Hall/CRC, 2003.

[54] Chuanhai Liu. *Robit Regression: A Simple Robust Alternative to Logistic and Probit Regression*, pages 227–238. John Wiley & Sons, Ltd, 2005.

[55] David W. Hosmer and Stanley Lemeshow. *Applied logistic regression (Wiley Series in probability and statistics)*. Wiley-Interscience Publication, 2 edition, 2000.

[56] Havard Rue, Sara Martino, and Nicolas Chopin. Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations. *Journal of the Royal Statistical Society Series B*, 71(2):319–392, 2009.

[57] J. Gentle. *Solutions that Minimize Other Norms of the Residuals*. New York: Springer, 2007.

[58] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. *Section 7.9.1 Importance Sampling*. New York: Cambridge University Press, 2007.

[59] Sara Martino and Nicolas Chopin. Implementing approximate Bayesian inference for latent Gaussian models using integrated nested Laplace approximations: A manual for the inla-program. Technical report, 2008.

[60] Ralph E. Steuer, Jurgen Kurths, Carsten O. Daub, Janko Weise, and Joachim Selbig. The mutual information: Detecting and evaluating dependencies between variables. In *ECCB*, pages 231–240, 2002.

[61] Aleksandar Lazarevic and Vipin Kumar. Feature bagging for outlier detection. In *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, KDD '05, pages 157–166, New York, NY, USA, 2005. ACM.

[62] Kaustav Das and Jeff Schneider. Detecting anomalous records in categorical datasets. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '07, pages 220–229, New York, NY, USA, 2007. ACM.

[63] A. Frank and A. Asuncion. UCI Machine Learning Repository. http://archive.ics.uci.edu/ml/, 2010.

[64] Zengyou He, Xiaofei Xu, Joshua Zhexue Huang, and Shengchun Deng. Fp-outlier: Frequent pattern based outlier detection. *Computational Science Information System*, 2(1):103–118, 2005.

[65] Shu Wu and Shengrui Wang. Information-theoretic outlier detection for large-scale categorical data. *IEEE Transactions on Knowledge and Data Engineering*, 25(3):589–602, 2013.

[66] Yen-Cheng Lu, Chih-Wei Wu, Chang-Tien Lu, and Alexander Lerch. An unsupervised approach to anomaly detection in music datasets. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '16, page 749–752, New York, NY, USA, 2016. Association for Computing Machinery.

[67] Yen-Cheng Lu, Chih-Wei Wu, Alexander Lerch, and Chang-Tien Lu. Automatic Outlier Detection in Music Genre Datasets. In *Proceedings of the 17th International Society for Music Information Retrieval Conference*, pages 101–107, New York City, United States, August 2016. ISMIR.

[68] Mikhail Atallah, Wojciech Szpankowski, and Robert Gwadera. Detection of significant sets of episodes in event sequences. In *Data Mining, 2004. ICDM '04. Fourth IEEE International Conference on*, pages 3–10, Nov 2004.

[69] Patrick L. Brockett, Xiaohua Xia, and Richard A. Derrig. Using kohonen's self-organizing feature map to uncover automobile bodily injury claims fraud. *The Journal of Risk and Insurance*, 65(2):245–274, 1998.

[70] Eun Park, Shawn Turner, and Clifford Spiegelman. Empirical approaches to outlier detection in intelligent transportation systems data. *Transportation Research Record: Journal of the Transportation Research Board*, 1840:21–30, 2003.

[71] Michael R. Smith and Tony Martinez. Improving classification accuracy by identifying and removing instances that should be misclassified. In *Proceedings of International Joint Conference on Neural Networks*, pages 2690–2697, 2011.

[72] Markus Schedl, Emilia Gómez, and Julián Urbano. Music Information Retrieval: Recent Developments and Applications. *Foundations and Trends in Information Retrieval*, 8:127–261, 2014.

[73] Sridhar Ramaswamy, Rajeev Rastogi, and Kyuseok Shim. Efficient algorithms for mining outliers from large data sets. *SIGMOD Record*, 29(2):427–438, May 2000.

[74] Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, and Jörg Sander. Lof: identifying density-based local outliers. *SIGMOD Record*, 29(2):93–104, May 2000.

[75] Cláudia Pascoal, M. Rosário Oliveira, António Pacheco, and Rui Valadas. Detection of outliers using robust principal component analysis: A simulation study. In Christian Borgelt, Gil González-Rodríguez, Wolfgang Trutschnig, María Asunción Lubiano, María Ángeles Gil, Przemysław Grzegorzewski, and Olgierd Hryniewicz, editors, *Combining Soft Computing and Statistical Methods in Data Analysis*, pages 499–507. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[76] Emmanuel J. Candès, Xiaodong Li, Yi Ma, and John Wright. Robust principal component analysis? *Journal of the ACM*, 58(3):11:1–11:37, June 2011.

[77] Arthur Flexer, Elias Pampalk Gerhard, and Gerhard Widmer. Novelty detection based on spectral similarity of songs arthur flexer. In *in Proc. of the 6 th Int. Symposium on Music Information Retrieval*. Ms, 2005.

[78] Lars Kai Hansen, Tue Lehn-Schiøler, Kaare Brandt Petersen, Jeronimo Arenas-Garcia, Jan Larsen, and Søren Holdt Jensen. Learning and clean-up in a large scale music database. In *European Signal Processing Conference (EUSIPCO)*, pages 946–950, 2007.

[79] Zhouyu Fu, Guojun Lu, Kai Ming Ting, and Dengsheng Zhang. A Survey of Audio-Based Music Classification and Annotation. *IEEE Transactions on Multimedia*, 13(2):303–319, April 2011.

[80] Alexander Lerch. *An Introduction to Audio Content Analysis: Applications in Signal Processing and Music Informatics*. John Wiley and Sons, 2012.

[81] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM Computer Survey*, 41(3):15:1–15:58, July 2009.

[82] Raheda Smith, Alan Bivens, Mark Embrechts, Chandrika Palagiri, and Boleslaw Szymanski. Clustering approaches for anomaly based intrusion detection. In *Proceedings of intelligent engineering systems through artificial neural networks*, 2002.

[83] Eleazar Eskin, Andrew Arnold, Michael Prerau, Leonid Portnoy, and Sal Stolfo. A geometric framework for unsupervised anomaly detection. In Daniel Barbará and Sushil Jajodia, editors, *Applications of Data Mining in Computer Security*, pages 77–101. Springer US, Boston, MA, 2002.

[84] Bernhard Schölkopf, John C. Platt, John C. Shawe-Taylor, Alex J. Smola, and Robert C. Williamson. Estimating the support of a high-dimensional distribution. *Neural Comput.*, 13(7):1443–1471, July 2001.

[85] David E. Tyler. Robust statistics: Theory and methods. *Journal of the American Statistical Association*, 103:888–889, 2008.

[86] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[87] Christian P. Robert and George Casella. *Markov Chains*, pages 205–265. Springer New York, New York, NY, 2004.

[88] Peter Müller. Alternatives to the gibbs sampling scheme. Technical report, Institute of Statistics and Decision Sciences, Duke University, 1993.

[89] Bob L. Sturm. Music genre recognition with risk and rejection. In *2013 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6, July 2013.

[90] W. Yu and N. Wang. Research on credit card fraud detection model based on distance sum. In *2009 International Joint Conference on Artificial Intelligence*, pages 353–356, 2009.

[91] Jian Bo Yang, Minh Nhut Nguyen, Phyo Phyo San, Xiao Li Li, and Shonali Krishnaswamy. Deep convolutional neural networks on multichannel time series for human activity recognition. In *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, page 3995–4001. AAAI Press, 2015.

[92] Matthew Dixon, Diego Klabjan, and Jin Hoon Bang. Classification-based financial markets prediction using deep neural networks. *Algorithmic Finance*, 6(3-4):67–77, January 2017.

[93] László A. Jeni, András Lőrincz, Zoltán Szabó, Jeffrey F. Cohn, and Takeo Kanade. Spatio-temporal event classification using time-series kernel based structured sparsity. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 135–150, Cham, 2014. Springer International Publishing.

[94] S. M. Bendre and B. K. Kale. Masking effect on tests for outliers in exponential models. *Journal of the American Statistical Association*, 80(392):1020–1025, 1985.

[95] M. Gupta, J. Gao, C. C. Aggarwal, and J. Han. Outlier detection for temporal data: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 26(9):2250–2267, 2014.

[96] Eleazar Eskin, Andrew Arnold, Michael Prerau, Leonid Portnoy, and Salvatore Stolfo. A geometric framework for unsupervised anomaly detection: Detecting intrusions in unlabeled data. *Applications of Data Mining in Computer Security*, 6, 02 2002.

[97] Bo Gao, Hui-Ye Ma, and Yu-Hang Yang. Hmms (hidden markov models) based on anomaly intrusion detection method. In *Proceedings. International Conference on Machine Learning and Cybernetics*, volume 1, pages 381–385 vol.1, 2002.

[98] M. Atallah, W. Szpankowski, and R. Gwadera. Detection of significant sets of episodes in event sequences. In *Fourth IEEE International Conference on Data Mining (ICDM'04)*, pages 3–10, 2004.

[99] Rohit Kate. Using dynamic time warping distances as features for improved time series classification. *Data Mining and Knowledge Discovery*, 30, 05 2015.

[100] Jon Hills, Jason Lines, Edgaras Baranauskas, James Mapp, and Anthony Bagnall. Classification of time series by shapelet transformation. *Data Mining and Knowledge Discovery*, 28, 05 2013.

[101] J. Lines, S. Taylor, and A. Bagnall. Hive-cote: The hierarchical vote collective of transformation-based ensembles for time series classification. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pages 1041–1046, 2016.

[102] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[103] T. Young, D. Hazarika, S. Poria, and E. Cambria. Recent trends in deep learning based natural language processing [review article]. *IEEE Computational Intelligence Magazine*, 13(3):55–75, 2018.

[104] Bendong Zhao, Huanzhang Lu, Shangfeng Chen, Junliang Liu, and Dongya Wu. Convolutional neural networks for time series classification. *Journal of Systems Engineering and Electronics*, 28(1):162–169, 2017.

[105] Zachary Chase Lipton, David C. Kale, Charles Elkan, and Randall C. Wetzel. Learning to diagnose with LSTM recurrent neural networks. In Yoshua Bengio and Yann LeCun, editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016.

[106] Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, and Pierre-Alain Muller. Deep learning for time series classification: a review. *Data Mining and Knowledge Discovery*, 03 2019.

[107] Yann Lecun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, pages 2278–2324, 1998.

[108] Yu hsin Chen, Ignacio Lopez Moreno, Tara Sainath, Mirkó Visontai, Raziel Alvarez, and Carolina Parada. Locally-connected and convolutional neural networks for small footprint speaker recognition. In *Interspeech*, 2015.

[109] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *NIPS Deep Learning workshop,2014*.

[110] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter. Audio set: An ontology and human-labeled dataset for audio events. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 776–780, 2017.

[111] N. Hammami and M. Bedda. Improved tree model for arabic speech recognition. In *2010 3rd International Conference on Computer Science and Information Technology*, volume 5, pages 521–526, 2010.

[112] Fevzi Alimoglu, Yrd Doc, Dr. Ethem Alpaydin, Doc Dr, and Yagmur Denizhan. Combining multiple classifiers for pen-based handwritten digit recognition, 1996.

[113] H. Hamooni and A. Mueen. Dual-domain hierarchical classification of phonetic time series. In *2014 IEEE International Conference on Data Mining*, pages 160–169, 2014.

Appendix A

Appendix

A.1 Equations for updating β

We state the brief derivation and result for updating β in the variational EM framework for each data type in this appendix.

Binary:
Since setting the above Gradient to zero has no analytical solution. This update is accomplished by solving an inner optimization problem on β_p, s.t.

$$\beta_p^{(new)} = \arg\max_{\beta_p} \left(\ln p(\beta_p) + \sum_{n=1}^{N} \ln p(Y_{np}|\eta_{np}\hat{\nu}) \right) \quad \text{(A.1)}$$

with the following gradient and Hessian:

$$\begin{aligned} &\nabla_{\beta_p} \mathcal{Q}^{(\beta)}(\theta, \theta^{old}) \\ &= \sum_{n=1}^{N} X_n^T \left(Y_{np} - \frac{1}{1 + \exp\left[-(\eta_{np}\hat{\nu})\right]} \right) \\ &\qquad - \Sigma_{\beta_p}^{-1}(\beta_p - \mu_{\beta_p}) \end{aligned} \quad \text{(A.2)}$$

$$\nabla\nabla_{\beta_p}\mathcal{Q}^{(\beta)}(\theta,\theta^{old}) = -\Sigma_{\beta_p}^{-1} - \sum_{n=1}^{N} X_n^T X_n \left(\frac{\exp\{-(\eta_{np}\hat{\nu})\}}{(1+\exp\{-(\eta_{np}\hat{\nu})\})^2}\right) \quad \text{(A.3)}$$

For convenience, we use $\eta_{np}\hat{\nu}$ to denote $X_n\beta_p + \hat{\omega}_{np} + \hat{\varepsilon}_{np}$ in this equation and the later equations.

Count:

Similar to the binary type, maximizing count likelihood has no analytical solution. Therefore, β_p in count type is updated with an inner optimization with the gradient and Hessian below:

$$\nabla_{\beta_p}\mathcal{Q}^{(\beta)}(\theta,\theta^{old}) = -\Sigma_{\beta_p}^{-1}(\beta_p - \mu_{\beta_p}) + \sum_{n=1}^{N} X_n^T \left(\frac{Y_{np}}{\eta_{np}\hat{\nu}} - 1\right) \quad \text{(A.4)}$$

$$\nabla\nabla_{\beta_p}\mathcal{Q}^{(\beta)}(\theta,\theta^{old}) = -\Sigma_{\beta_p}^{-1} - \sum_{n=1}^{N} X_n^T X_n \left(\frac{Y_{np}}{(\eta_{np}\hat{\nu})^2}\right) \quad \text{(A.5)}$$

Categorical:

The categorical likelihood is analytically intractable as well, the gradient and Hessian are:

$$\nabla_{\beta_{pk}}\mathcal{Q}^{(\beta)}(\theta,\theta^{old}) = \sum_{n=1}^{N} X_n \left[Y_{npk} - \frac{\exp\{\eta_{npk}\hat{\nu}\}}{1+\sum_{l=1}^{K-1}\exp\{\eta_{npl}\hat{\nu}\}}\right] - \Sigma_{\beta_p}^{-1}(\beta_{pk} - \mu_{pk}) \quad \text{(A.6)}$$

$$\nabla\nabla_{\beta_{pk}}\mathcal{Q}^{(\beta)}(\theta,\theta^{old})$$
$$= -\sum_{i=1}^{N} x_i x_i^T \left[\frac{e^{\eta_{npk}\hat{\nu}}[1+(\sum_{l=1}^{D-1} e^{\eta_{npl}\hat{\nu}}) - e^{\eta_{npk}\hat{\nu}}]}{(1+\sum_{l=1}^{K-1} e^{\{\eta_{npl}\hat{\nu}\}})^2}\right]$$
$$- \Sigma_{\beta_p}^{-1} \quad \text{(A.7)}$$