AUTOMATIC OUTLIER DETECTION IN MUSIC GENRE DATASETS

Yen-Cheng Lu[1] **Chih-Wei Wu**[2] **Chang-Tien Lu**[1] **Alexander Lerch**[2]

[1] Department of Computer Science, Virginia Tech, USA
[2] Center for Music Technology, Georgia Institute of Technology, USA
kevinlu@vt.edu, cwu307@gatech.edu, ctlu@vt.edu, alexander.lerch@gatech.edu

ABSTRACT

Outlier detection, also known as anomaly detection, is an important topic that has been studied for decades. An outlier detection system is able to identify anomalies in a dataset and thus improve data integrity by removing the detected outliers. It has been successfully applied to different types of data in various fields such as cyber-security, finance, and transportation. In the field of Music Information Retrieval (MIR), however, the number of related studies is small. In this paper, we introduce different state-of-the-art outlier detection techniques and evaluate their viability in the context of music datasets. More specifically, we present a comparative study of 6 outlier detection algorithms applied to a Music Genre Recognition (MGR) dataset. It is determined how well algorithms can identify mislabeled or corrupted files, and how much the quality of the dataset can be improved. Results indicate that state-of-the-art anomaly detection systems have problems identifying anomalies in MGR datasets reliably.

1. INTRODUCTION

With the advance of computer-centric technologies in the last few decades, various types of digital data are being generated at an unprecedented rate. To account for this drastic growth in digital data, exploiting its (hidden) information with both efficiency and accuracy became an active research field generally known as Data Mining.

Outlier detection, being one of the most frequently studied topics in Data Mining, is a task that aims to identify abnormal data points in the investigated dataset. Generally speaking, an outlier often refers to the instance that does not conform to the expected behavior and should be highlighted. For example, in a security surveillance system an outlier could be the intruder, whereas in credit card records, an outlier could be a fraud transaction.

Many algorithms have been proposed to identify outliers in different types of data, and they have been proven successful in fields such as cyber-security [1], finance [4], and transportation [17]. Outlier detection techniques can also be used as a pre-processing step to remove anomalous data. In the work of Smith and Martinez [24], a set of outlier detection methods are used to remove anomalies from the dataset, followed by several widely-used classification methods in order to compare the performance before and after outlier removal. The result indicates that removing outliers could lead to statistically significant improvements in the training quality as well as the classification accuracy for most of the cases.

Music datasets offer similar challenges to researchers in the field of MIR. Schedl et al. point out that many MIR studies require different datasets and annotations depending on the task [22]. However, since the annotation of music data is complex and subjective, the quality of the annotations created by human experts varies from dataset to dataset. This inaccuracy may potentially introduce errors to the system and decrease the resulting performance.

One MIR task known for this issue is Music Genre Recognition (MGR). According to Sturm [26], the most frequently used dataset in MGR is GTZAN [29], and many of the existing systems are evaluated based on their performance on this dataset. Sturm points out that this dataset contains corrupted files, repeated clips, and misclassified genre labels. These are undesirable for the proper training and testing of a MGR system.

To address the problem of identifying such anomalies in music datasets, an investigation into existing outlier detection algorithms is a good starting point. The goal of this paper is to assess the viability of state-of-the-art outlier detection methods in the context of music data. The contribution of this paper can be summarized as follows: first, this early stage investigation provides a systematic assessment of different outlier detection algorithms applied to a music dataset. Second, the use of standard audio features reveals the capability as well as the limitations of this feature representation for outlier detection. Third, we provide insights and future directions for related studies.

This paper is structured as follows. In Sect. 2, the related work of outlier detection in music data is summarized. The methods used in this paper, such as feature extraction and different outlier detection algorithms, are described in Sect. 3. Section 4 presents the results and discusses the experiments. Finally, the conclusion and future work are given in Sect. 5.

2. RELATED WORK

Outlier detection methods are typically categorized into five groups: (a) *distance-based* [14, 19], (b) *density-based* [3, 16], (c) *cluster-based* [30], (d) *classification-based* [8, 21], and (e) *statistics-based* [5, 6, 13, 18, 20, 28] methods.

The first group (*distance-based*), proposed by Knorr et al. [14], computes the distances between samples and detects outliers by setting a distance threshold. Methods in this category are usually straightforward and efficient, but the accuracy is compromised when the data is sparse or unevenly distributed. The basic idea was extended by combining the distance criterion with the k-nearest neighbor (KNN) based method [19], which adapts the distance threshold by the k-nearest neighboring distances.

The second group (*density-based*) estimates the local densities around the points of interest in order to determine the outliers. Different variations use different methods to determine the local density, for example, the local outlier factor (LOF) [3] and the local correlation integral (LOCI) [16]. These approaches are popular and have been widely used in different fields.

The third group (*clustering-based*), as proposed in [30], first applies a clustering algorithm to the data, and then labels the wrongly clustered instances as outliers.

The fourth group (*classification-based*) assumes that the designation of anomalies can be learned by a classification algorithm. Here, classification models are applied to classify the instances into inliers and outliers. This is exemplified by Das et al. [8] with a one-class Support Vector Machine (SVM) based approach and Roth [21] with Kernel Fisher Discriminants.

The fifth group (*statistics-based*) assumes the data has a specific underlying distribution, and the outliers can be identified by finding the instances with low probability densities. A number of works apply the similar concept with variations, including techniques based on the robust Mahalanobis distance [20], direction density ratio estimation [13], and minimum covariance determinant estimator [6]. One of the main challenges of these approaches is the reduction of masking and swamping effects: outliers can bias the estimation of distribution parameters, yielding biased probability densities. This effect could result in a biased detector identifying normal instances as outliers, and outliers as normal, respectively. Recent advances have generally focused on applying robust statistics to outlier detection [5, 18, 28]. This is usually achieved by adopting a robust inference technique to keep the model unbiased from outliers in order to capture the normal pattern correctly.

Although the above mentioned approaches have been applied to different types of data, the number of studies on music datasets is relatively small. Flexer et al. [10] proposed a novelty detection approach to automatically identify new or unknown instances that are not covered by the training data. The method was tested on a MGR dataset with 22 genres and was shown to be effective in a cross-validation setting. However, in real-world scenarios, the outliers are usually hidden in the dataset, and an outlier-free training dataset may not be available. As a result, the proposed method might not be directly applicable to other music datasets. Hansen et al. [12] proposed the automatic detection of anomalies with a supervised method based on parzen-window and kernel density estimation. The proposed algorithm was evaluated on a 4-class MGR dataset, which consisted of audio data recorded from radio stations. A commonly used set of audio features, the Mel Frequency Cepstral Coefficients (MFCCs), was extracted to represent the music signals. This approach, nevertheless, has two underlying problems. First, the dataset used for evaluation does not have a ground truth agreed on by human experts. Second, while MFCCs are known to be useful in a multitude of MIR tasks, they might not be sufficient to represent music signals for outlier detection tasks.

To address these issues, two approaches have been taken in this paper: First, for evaluation, a commonly-used MGR dataset with reliable ground truth is used. In Sturm's analysis [26], a set of outliers (i.e., repeated, distorted, and mislabeled music clips) were identified manually in the popular GTZAN [29] dataset. This analysis provides a solid ground for the evaluation of an anomaly detection system in the MGR dataset. Second, we extend the set of descriptors for the music data. In addition to the MFCCs, audio features that are commonly used in MIR tasks are also extracted in order to evaluate the compatibility of current audio features with the existing outlier detection methods.

3. METHOD

3.1 Feature Extraction

Feature extraction is an important stage that transforms an audio signal into a vector-based representation for further data analysis. In an early study of automatic music genre classification, Tzanetakis and Cook proposed three feature sets that characterized any given music signal based on its timbral texture, rhythmic content and pitch content [29]. These features have shown their usefulness in music genre classification, and have been used in many music-related tasks. Although many studies presented more sophisticated features (e.g., [11]) with higher classification accuracy on the GTZAN dataset, the original set of features still seem to provide a good starting point for representing music data. Therefore, a set of baseline features based on Tzanetakis and Cook's features [29] is extracted to allow for easier comparison with prior work. The extracted features can be divided into three categories: spectral, temporal and rhythmic. All of the features are extracted using a block-wise analysis method. To begin with, the audio signal is down-mixed to a mono signal. Next, a Short Time Fourier Transform (STFT) is performed using a block size of 23 ms and a hop size of 11 ms with a Hann window in order to obtain the time-frequency representation. Finally, different instantaneous features are extracted from every block. Spectral features are computed using the spectrum of each block. Temporal features are computed from the time domain signal of each block directly. The rhythmic features are extracted from the beat histogram of the entire time domain signal. The extracted features are (for the details of the implementations, see [15]):

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 of the features are aggregated into texture vectors following the standard procedure as mentioned in [29]; the length of the current texture block is 0.743 s. The mean and standard deviation of the feature vectors within this time span will be computed to create a new feature vector. Finally, all the texture blocks will be summarized again by the mean and standard deviation of these blocks, generating one feature vector to represent each individual recording in the dataset.

3.2 Outlier Detection Methods

3.2.1 Problem Definition

Given N music clips that have been converted into 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 belong to one of the M genres (i.e., $Y_n \in \{C_1, ..., C_M\}$), the objective is to find the indices of the abnormal instances which have an incorrect label Y_n.

For this study, we choose 6 well-known outlier detection methods from different categories as introduced in Sect. 2 and compare their performances on a MGR dataset. The methods are described in details in the following sections:

3.2.2 Method 1: Clustering

Clustering is a *cluster-based* approach as described in Sect. 2. In our implementation of this method, we apply k-means to cluster the data into 10 groups. Based on the assumption that normal data are near the cluster centroids while the outliers are not [7, 25], the anomalous score of a given instance is defined by the distance between the point and the centroid of the majority within the same class.

3.2.3 Method 2: KNN

KNN method is a *distance-based* approach that typically defines the anomalous score of each instance by its distance to the k nearest neighbors [9]. It can be expressed in the following equation:

$$k\text{–}distance(P) = d(P, knn(P)) \tag{1}$$

where knn is the function that returns the k-th nearest neighbor of a point P, and d is the function that calculates the distance between two points. Finally, we may compute the outlier score as:

$$\frac{1}{k} \sum_{p \in neighbors_k(P)} k\text{–}distance(p) \tag{2}$$

Setting k to a larger number usually results in a model more robust against outliers. When k is small, the anomalous score given by this method may be biased by a small group of outliers. In our implementation of this method, we apply $k = 6$ in order to maintain a balance between robustness and efficiency.

3.2.4 Method 3: Local Outlier Factor

The Local Outlier Factor (LOF) [3] is a *density-based* approach that extends the KNN method with a calculation of the local densities of the instances. It is one of the most popular anomaly detection methods. It starts with the definition of k-reachability distance:

$$k\text{–}reachDist(P, O) = max(k\text{–}distance(P), d(O, P)) \tag{3}$$

This represents the distance from O to P, but not less than the $k\text{–}distance$ of P. The local reachability density of a given sample is defined by the inverse of the average local rechability distances of k-nearest neighbors:

$$lrd(P) = 1/\left(\frac{\sum_{P_0 \in neighbors_k(P)} k\text{–}reachDist(P, P_0)}{|neighbors_k(P)|}\right) \tag{4}$$

Finally, the lof calculates the average ratio of the local reachability densities of the k-nearest neighbors against the point P:

$$lof(P) = \frac{\sum_{P_0 \in neighbors_k(P)} lrd(P_0)}{lrd(P)|neighbors_k(P)|} \tag{5}$$

In a dataset that is densely distributed, a point may have shorter average distance to its neighbors, and vice versa. Since LOF uses the ratio instead of the distance as the outlier score, it is able to detect outliers in clusters with different densities.

3.2.5 Method 4: One-Class SVM

The One-Class SVM [23] is a *classification-based* approach that identifies outliers with a binary classifier. Given a genre $m \in \{1, ..., M\}$, every sample in m can be classified as in-class or off-class, and the off-class instances are most likely to be the outliers. A One-Class SVM solves the following quadratic programming problem:

$$\begin{aligned} \min_{w, \xi_i, \rho} \quad & \tfrac{1}{2}||w||^2 + \tfrac{1}{\nu N} \textstyle\sum_i \xi_i - \rho \\ \text{subject to} \quad & (w\Phi(x_i)) \geq \rho - \xi_i, i = 1...N \\ & \xi_i \geq 0, i = 1...N \end{aligned} \tag{6}$$

where ξ, w, and ρ are the parameters to construct the separation hyperplane, ν is a parameter that serves as the upper bound fraction of outliers and a lower bound fraction of samples used as support vectors, and Φ is a function that maps the data into an inner product space such that the projected data can be modeled by some kernels such as a Gaussian Radial Basis Function (RBF). By optimizing the above objective function, a hyperplane is then created to separate the in-class instances from the outliers. In the experiment, we construct a One-Class SVM for each of the genres, and identify the music clips that are classified as off-class instances as outliers.

3.2.6 Method 5: Robust PCA

Robust PCA [5] is a *statistics-based* approach that considers the problem of 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 [5]:

$$minimize||L||_* + \lambda||S||_1 \quad (7)$$

$$\text{subject to } L + S = X \quad (8)$$

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 consisting of mostly zero entries with a few non-zero entries being the outliers. In the experiment, we apply this method to the music data and calculate the sparse matrices for every genre. Next, we normalize the features using a standard Z-score normalization process, and identify the outliers by finding the instances with a maximum sparse matrix value that is 3 times greater than the unity standard deviation.

3.2.7 Method 6: Robust Categorical Regression

Robust Categorial Regression (RCR) is another *statistics-based* method that identifies outliers based on a regression model. First, we formulate the relation of Y and X based on a linear input-output assumption:

$$g(Y) = X\beta + \varepsilon, \quad (9)$$

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, i.e., $\ln\left(P(Y_n = C_m)/P(Y_n = C_M)\right) = X_n\beta_m + \varepsilon_{nm}$. Since the probabilities of the categories will sum up to one, we can derive the following modeling equation:

$$P(Y_n = C_m) = \frac{\exp\left\{X_n\beta_m + \varepsilon_{nm}\right\}}{1 + \sum_{l=1}^{M-1}\exp\left\{X_n\beta_l + \varepsilon_{nl}\right\}} \quad (10)$$

and

$$P(Y_n = C_M) = \frac{1}{1 + \sum_{l=1}^{M-1}\exp\left\{X_n\beta_l + \varepsilon_{nl}\right\}} \quad (11)$$

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.

The robust version of categorical regression applies a heavy-tailed distribution, which is a zero-mean Student-t distribution to capture the error effect caused by outliers.

The solution to this regression model is approximated with a variational Expectation-Maximization (EM) algorithm [2]. Once converged, the errors of the instances are expected to be absorbed in the ε variables. Finally, the outliers can be identified by finding the instances with ε that is 3 times greater than the unity standard deviation.

4. EXPERIMENT

4.1 Experiment Setup

To evaluate the state-of-the-art outlier detection methods as described in Sect. 3.2, different experiments are conducted on the well-known GTZAN dataset [29]. This dataset consists of 10 music genres (i.e., blues, classical, country, disco, hip-hop, jazz, metal, pop, reggae, and rock), with each genre containing 100 audio tracks; each track is a 30-second long excerpt from a complete mixture of music. For each method, two sets of experiments are conducted.

In the first set of experiments, we use a purified GTZAN dataset, which excludes the conspicuous misclassified and jitter music clips reported in [26]. This setup simulates the best case scenario, where the dataset is clean and all genres are well separated in the feature space. The results can serve as a sanity check of all the methods. Two types of injection experiments are conducted on this purified dataset, namely label injection and noise injection. The label injection process is performed by randomly choosing 5% of instances, and swapping their genre labels to create outliers. In this experiment, two sets of features are used to represent the music data, one is the full feature set as described in Sect. 3.1, and the other is the baseline feature set using only 13 MFCCs as reported in the work of Hansen et al. [12] for comparison. The noise injection process is performed by randomly choosing 5% of instances in the data, and shifting 20% of their feature values by 5 times the standard deviation. This experiment uses the full feature set to test the methods' capability of detecting corrupted data. For each of the experiments above, we generate 10 random realizations and report the averaged evaluation results.

In the second set of experiments, we apply all the methods to the full GTZAN dataset directly, and the identified outliers are compared with the list of conspicuous genre labels and the obviously corrupted clips (*Hip-hop (38), Pop (37), Reggae (86)*) reported in [26]. This experiment provides the real-world scenario, in which case the outlier detection should find the outliers identified by human experts.

All of the experiments use the same metrics for the performance measurements, which include the standard calculation of Precision, Recall, F-measure, and the Area Under ROC Curves (AUC).

4.2 Experiment Results

The results of the first set of experiments, evaluating the performance of the methods on detecting injected misclassification labels with full features, are shown in Table 1. With F-measures in the range from 0.1–0.57, the results do not have high reliability but are usable for some methods. The *Robust Categorical Regression* approach outperforms the other algorithms. Since RCR explicitly models the input-output relationship between the features and the labels, it fits the data better compared to the other methods. Surprisingly, the simple methods such as *Clustering* and *KNN* also perform relatively well in terms of AUC, and they outperform the more sophisticated approaches such as *LOF*

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 1. Average Detection Rate 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 2. Average Detection Rate comparison of label injection with MFCCs only

and *One-Class SVM*. One possible explanation is that the label injection datasets contain outliers generated by swapping dissimilar genres, e.g., swapping the label from Jazz to Metal. As a result, the decision boundaries of *LOF* and *One-Class SVM* might be biased towards the extreme values and perform poorly. On the other hand, the simple methods such as *Clustering* and *KNN*, which are based on Euclidean distance, were able to identify these instances without being biased. Generally speaking, the *statistics-based* approaches, such as the robust statistics based methods *RPCA* and *Robust Categorical Regression*, perform better on the label injection datasets.

The results of the same experiments with only MFCCs as features are shown in Table 2. In general, the performance drops drastically. This result implies that MFCCs might not be representative enough for the outlier detection task.

Table 3 shows the results for the noise injection experiment. It can be observed that density and distance methods, such as *CLUS*, *KNN*, and *LOF*, have better results on detecting corrected data. The main distinction of this kind of outlier is that the abnormal behavior is explicitly shown in the feature space instead of implicitly embedded in the relationship between genre labels and the features. Therefore, the methods that directly detect outliers in the feature space tend to outperform the 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, and aim to detect the misclassified music clips reported by Sturm [26]. The experiment result is shown in Table 4. Based on these metrics, none of these methods are able to detect the anomalies with high accuracy. Compared

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. Average Detection Rate 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 4. Detection Rate comparison on GTZAN detecting Sturm's anomalies with full features

to the other methods, *SVM* and *RCR* present AUCs that are relatively higher, however, the Precision, Recall and F-measures are still too low to be applicable in real-world scenarios. The *One-Class SVM* method performs better in this experiment than it does in the previous experiment. We speculated that in the case of injection, the model is biased by the extreme values introduced by the injected outliers. In the real-world scenario, however, the differences between the outliers and the non-outliers are relatively subtle. When *One-Class SVM* expands its in-class region moderately, it learns a better decision boundary. Therefore, it has a better capability of detecting the outliers.

It can be observed that both *statistics-based* approaches, *RPCA* and *RCR*, do not perform well compared to the results of the previous experiment. Since these methods are good at capturing extreme values and prevent the model from being biased by the outliers, they are relatively weak in differentiating subtle differences in the feature space. Therefore, the resulting performances are not ideal.

4.3 Discussion

To further reveal the relationship between different methods and outliers from different genres, we list the distribution of top 20 true and false outliers ranked by the anomalous scores of different methods as well as the true distribution reported by Sturm [26]. The results are shown in Table 5. Interestingly, majority of the approaches have most of the true outliers in *Disco* and *Reggae* except the *One-Class SVM*. For the *One-Class SVM*, its top 20 includes 14 metal outliers, which are barely detected by the other methods. More specifically, the *One-Class SVM* had a high precision of 14/26 in the *Metal* genre. Since most of the true outliers in the *Metal* genre can be categorized to punk rock according to the definition on the online music library, [1] they could exhibit similar features with subtle differences in the feature space, and they are still detected by the *One-Class SVM*. In *Reggae*, there is a jitter music clip which presents extreme values in the feature space, along with the other outliers. For the *One-Class SVM* in the context of *Reggae*, however, only the jitter instance is captured while the other outliers are missing. These two observations confirm that *One-Class SVM* is especially good at distinguishing the outliers that have subtle differences, and could be easily biased by the outliers with extreme values.

Three of methods have about 10 of the top 20 false outliers in *Pop*. This may due to the variety of *Pop* music in the dataset. For example, although *Pop (12) - Aretha Franklin, Celine Dion, Mariah Carey, Shania Twain, and Gloria Es-*

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

	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 5. Distribution of the Top 20 Ranked True Outliers/False Outliers among Methods.

tefan "You Make Me Feel Like A Natural Woman" is not identified by the expert as an outlier, it can be argued to be *Soul* music. By the nature of the *One-Class SVM*, this clip is also ranked at the top by its anomalous score. Another interesting observation is that four methods have about 5 *jazz* instances in the top 20 false outliers. Although *jazz* music has strong characteristics and can be easily distinguished by humans, it shows the most significant average variance on its features comparing with other genres. Thus, the methods that calculate the Euclidean distances such as *Clustering*, *KNN*, and *LOF*, and the approaches that absorb variances as errors such as *RPCA* could report more false outliers in this circumstance. We also noticed that *RCR* includes 10 *Rock* false outliers in its top 20. This may because it models the input-output relationship among all genres, and this global-view property thus causes the model mixing up *Rock* with other partially overlapping genres such as *Metal* and *Blues*.

To summarize, outlier detection in music data faces the following challenges compared to other types of data: first, due to the ambiguity in the genre definitions, some of the tracks can be considered as both outliers and non-outliers. This inconsistency may impact the training and testing results for both supervised and unsupervised approaches. In Sturm's [27] work, a risk model is proposed to model the loss of misclassification by the similarity of genres. Second, the music data has temporal dependencies. In the current framework, we aggregate the block-wise feature matrix into a single feature vector as it allows for the immediate use in the context of the state-of-the-art methods. This approach, however, does not keep the temporal changes of the music signals and potentially discards important information for identifying outliers with subtle differences. Third, the extracted low-level features might not be able to capture the high-level concept of music genre, therefore, it is difficult for the outlier detection algorithms to find the outliers agreed on by the human experts. Finally, the outliers are unevenly distributed among genres (e.g., *Metal* has 16 while *Blues* and *Classical* have none), and the data points are also distributed differently in the feature space in each genre. An approach or a specific parameter setting may perform well on some of the genres and fail on others.

5. CONCLUSION

In this paper, we have presented the application of outlier detection methods on a music dataset. Six state-of-the-art approaches have been investigated in the context of music genre recognition, and their performance is evaluated based on their capability of finding the outliers identified by human experts [26]. The results show that all of the methods fail to identify the outliers with reasonably high accuracy. This leaves room for future improvement in the automatic detection of outliers in music data. The experiment results also reveal the main challenges for outlier detection in music genre recognition: first, genre definitions are usually subjective and ambiguous. Second, the temporal dependencies of music need to be modeled. Third, the low-level audio features might not be able to capture the high-level concepts. These challenges may also generalize to other music datasets, and they should be further addressed in future work.

We identify possible directions for future work as: First, as shown in the experiments, a better feature representation should lead to a better performance for the majority of the methods. Therefore, to robustly isolate outliers, a better feature representation for outlier detection algorithms seems to be necessary. Second, since music data has temporal dependencies, the static approach in the current framework might not be feasible. An outlier detection method that can handle the temporal dependencies could potentially show improved performance Third, in the top 20 list for different methods, it is shown that different methods could be sensitive to different types of outliers. An ensemble approach that takes advantage of multiple methods might be considered in future studies.

With our results, we have shown that outlier detection in music datasets is still at a very early stage. To fully characterize a music signal, many challenges and questions still need to be answered. With current advances in feature design and feature learning, however, we expect significant progress to be made in the near future.

6. REFERENCES

[1] 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.

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

[3] 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.

[4] 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.

[5] 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.

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

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

[8] 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.

[9] 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.

[10] 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.

[11] 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.

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

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

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

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

[16] Spiros Papadimitriou, Hiroyuki Kitagawa, Christos Faloutsos, and Phillip B. Gibbons. Loci: fast outlier detection using the local correlation integral. In *Data Engineering, 2003. Proceedings. 19th International Conference on*, pages 315–326, March 2003.

[17] 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.

[18] Cludia Pascoal, M. Rosrio Oliveira, Antnio 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.

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

[20] 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.

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

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

[23] 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.

[24] 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.

[25] 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.

[26] 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.

[27] 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.

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

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

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