applied sciences

MDPI

Article

Evaluating Preprocessing Techniques for Unsupervised Mode Detection in Irish Traditional Music

Juan José Navarro-Cáceres *, Diego M. Jiménez-Bravo and María Navarro-Cáceres

Expert Systems and Applications Lab, Faculty of Science, University of Salamanca, Plaza de los Caídos s/n, 37008 Salamanca, Spain; dmjimenez@usal.es (D.M.J.-B.); maria90@usal.es (M.N.-C.)
* Correspondence: juanjo_810@usal.es

Abstract: Significant computational research has been dedicated to automatic key and mode detection in Western tonal music, particularly within the major and minor modes. However, limited research has focused on identifying alternative diatonic modes in traditional and folk music contexts. This paper addresses this gap by comparing the effectiveness of various preprocessing techniques in unsupervised machine learning for diatonic mode detection. Using a dataset of Irish folk music that incorporates diatonic modes such as Ionian, Dorian, Mixolydian, and Aeolian, we assess how different preprocessing approaches influence clustering accuracy and mode distinction. By examining multiple feature transformations and reductions, this study highlights the impact of preprocessing choices on clustering performance, aiming to optimize the unsupervised classification of diatonic modes in folk music traditions.

Keywords: mode detection; unsupervised learning; preprocessing; folk music

Academic Editor: Lamberto Tronchin

Received: 16 December 2024
Revised: 28 February 2025
Accepted: 10 March 2025
Published: 14 March 2025

Citation: Navarro-Cáceres, J.J.; Jiménez-Bravo, D.M.; Navarro-Cáceres, M. Evaluating Preprocessing Techniques for Unsupervised Mode Detection in Irish Traditional Music. *Appl. Sci.* **2025**, *15*, 3162. https://doi.org/10.3390/app15063162

1. Introduction

Traditional music is a pillar of cultural identity, reflecting the heritage and uniqueness of every community. However, the preservation of this type of music faces significant challenges due to its oral transmission tradition and the growing influence of commercial music and mass media in contemporary society. The potential loss of traditional music threatens the authenticity and cultural diversity of local communities. Digitalization is crucial for its preservation and global dissemination, contributing to a greater appreciation and understanding of musical traditions among international audiences.

One of the distinctive properties of traditional music is its mode, which has been studied by musicologists without digital support to gain a deeper understanding of cultural contexts [1–3]. However, when dealing with large datasets, this analysis requires the assistance of computational tools. Specifically, the application of Artificial Intelligence (AI) techniques to traditional music offers new opportunities to preserve and study its unique characteristics, with mode standing out as a distinctive and essential feature of this musical genre [4,5].

In music theory [2,3], a mode is a specific arrangement of whole and half steps within an octave, forming the basis of a melody or harmonic structure, i.e., it is the scale on which that melody is based. Modes are particularly relevant in folk music traditions, where melodies often do not conform to the harmonic expectations of major or minor keys. In Western folk music, modes such as Dorian, Mixolydian, and Aeolian are commonly used, each carrying a distinct tonal color and emotional quality.

A very similar concept is the *key* of a musical piece. A key in tonal music is defined by a tonic (central pitch) and a specific major or minor scale that establishes a hierarchical

relationship between notes and chords. In a given key, harmonic progressions follow a structured pattern, with certain chords (such as the dominant-to-tonic motion) reinforcing a strong sense of resolution. The key determines how tension and release function in the music, providing a framework for both melody and harmony.

While the concepts of mode and key share similarities (both describing pitch organization within a scale), they differ significantly in their musical functions and structural implications. A mode does not necessarily imply tonal harmony or a functional chord system. Instead, it primarily governs melodic characteristics, defining which notes are used and how they interact, without requiring harmonic resolution.

This distinction is crucial in folk music, where modal melodies often lack strong harmonic direction and may drift between tonal centers or avoid traditional cadences that reinforce a key. As a result, standard key detection algorithms, which rely on harmonic context and chord functions, may not be directly applicable to mode detection. Instead, mode classification requires analysis of melodic features, scale-degree distributions, and intervallic patterns rather than harmonic relationships.

In previous work, we developed a novel approach to mode detection that aims to identify the mode of a symbolically encoded modal music piece within the Irish folk music tradition [6]. This work demonstrates that detecting modes through machine learning algorithms is possible but also highlights the lack of knowledge regarding how to improve detection or determine the best approach. The field of AI algorithms, preprocessing techniques, classifiers, and paradigms is vast, requiring more specific research. Building on this foundation, the current paper presents a comparative study of various preprocessing techniques for unsupervised classification and their influence on the resulting performance. Specifically, we focus on three key aspects: exploring the impact of different feature representations, evaluating the effectiveness of alternative representation of data paradigms, and assessing the performance of a variety of unsupervised learning algorithms. For our work, we focus on Irish folk music using The Session, a broad database with more than 20,000 tagged files.

The remainder of this paper is organized as follows. Section 2 reviews the related work. Section 3 describes the dataset distribution and the algorithms applied to detect the mode. Section 4 reports and discusses the results of our experiments. Section 5 concludes the paper and suggests directions for future work.

2. Related Work

This section provides an overview of the key approaches employed in mode detection, encompassing different datasets, preprocessing techniques, and supervised and unsupervised learning paradigms for key detection.

2.1. Datasets and Encoding

To train the algorithms applied in this work, the data needed to be properly encoded and tagged with the mode and key. We were focused on the use of digital scores; therefore, the data needed to be encoded in MIDI, XML, MEI, or similar formats. It is important to note that the modal structure of folk songs can sometimes be blurred due to notation inconsistencies, which may affect their representation in digital datasets.

Different datasets have been used in the mentioned works. One of the most used is the Million Song Dataset, which is composed of the features of more than one million songs [7]. However, this dataset does not include the folk music we are seeking, i.e., traditional music that is separated by country of origin. The GTZAN dataset has been used in several works for key detection [8,9]. This dataset is composed of more than a thousand audio songs of different genres, such as rock, pop, jazz, and reggae. Although the dataset contains

the extracted features of the songs [10], it does not include folk music specifically. Finally, other datasets are composed of European folk music, including the Finnish Folk music archive [11] and The Session dataset, of which the latter is composed of Irish folk music [12].

In this work, we use The Session dataset because it contains the mode feature, whereas the Finnish folk music dataset would require a deeper musicological analysis before being included in our study. The Session is an online community dedicated to folk music, where members discuss relevant topics and share digitized scores encoded in ABC format, which can also be synthesized in MIDI. Therefore, since these are digital scores rather than recordings, they do not contain information about performers, interpretations, or contextual details. The only available item of metadata is the identity of the person who uploaded the scores. The dataset has been compiled by Irish folk music enthusiasts who are part of this community. ABC notation is a text-based format for encoding musical notes and structure and is simple to use for computational processing. Additionally, we have chosen The Session dataset because this research is part of a European project called EA-Digifolk, which focuses on the preservation of traditional music. In particular, one of the institutions involved in the project is the Irish Traditional Music Archive (ITMA), which specializes in Irish traditional music. The majority of the pieces that comprise The Session are dancing pieces. In particular, the distribution of genres is as follows: reel (35.91%), jig (25.42%), hornpipe (7.00%), strathspey (2.56%), polka (8.05%), barn dance (3.43%), slip jig (3.55%), march (2.78%), waltz (7.28%), slide (2.14%), mazurka (1.10%), and three–two (0.79%). These genres reflect the strong connection between Irish folk music and dance, as most traditional Irish music is composed as dancing pieces. More details on the processing of the dataset, including cleaning, preprocessing, and other relevant steps, are provided in Section 3.

2.2. Data Preprocessing

To obtain successful results, the data usually need to be preprocessed, extracting the features that provide useful information about the mode and key detection. Most of the previous works analyzed make use of Pitch Class Profile (PCP) extraction, based on the KK profiles by Krumhansl and Kessler [13,14]. Other profiles have been researched, such as the Temperley profiles [15] and the Zweiklang profiles [16]. Our work will use the KK profiles. This selection is based on a comprehensive review of literature, revealing its prevalence in various key detection papers [8,9,17,18].

In this technique, each mode can be represented by a pitch profile composed of the number of occurrences of the 12 notes of the musical temperament. The study by Krumhansl and Kessler was conducted for the Ionian and Aeolian modes. The main discovery in their research was the observation that both modes exhibited analogous characteristics for different pitches, with a consistent shift of the necessary number of semitones—the smallest interval in Western equal-temperament music—to align with the tonic center. This finding implies a shared tonal essence between these modes, emphasizing the transpositional relationship and the perceptual stability of certain pitches across different keys. Consequently, a representative histogram of a mode, composed of the frequency of each note, can be extracted from a musical piece. From this pitch profile, mode detection can be applied.

However, in recent years, embedding techniques have become central to various music information retrieval (MIR) tasks, including genre classification, mode detection, and recommendation systems [19–23]. Embeddings map musical features into continuous vector spaces, enabling more effective clustering and classification by capturing latent structure and relationships between musical elements.

Unlike traditional handcrafted features—such as pitch histograms, n-grams, or key profiles—embedding-based approaches leverage deep learning models (e.g., autoencoders, transformer networks, or contrastive learning) to extract compact and expressive feature

representations. These embeddings facilitate unsupervised and semi-supervised learning, making them particularly useful for tasks such as mode detection, where labeled data may be scarce.

Several embedding techniques have been explored in MIR [24–26], including the following:

- Spectrogram-based embeddings, which learn representations from time–frequency representations of audio signals;
- Symbolic music embeddings, which extract features from MIDI or ABC notation sequences, capturing melodic and harmonic patterns;
- Graph-based embeddings, which model relationships between musical elements, such as tonal connections or harmonic progressions.

In the context of mode detection, from our point of view, embedding approaches may enhance clustering performance by providing better separation between modes in high-dimensional spaces. However, one challenge is the interpretability of embeddings, as they do not always align with traditional musical descriptors. Furthermore, the choice of embedding model and training data can significantly impact performance, requiring careful selection and tuning.

2.3. Machine Learning Techniques in Mode Detection

Within the domain of music computing, there has been little research on mode detection. Most existing approaches focus on identifying the major and minor modes, which dominate Western classical and commercial music. In contrast, modes prevalent in modal folk music traditions—such as the Dorian, Phrygian, Lydian, Mixolydian, and Locrian modes—remain under-researched and need deeper investigation.

In contrast, key detection has been extensively studied in classical music. The concept of key or tonality plays a crucial role in this genre, characterized by a tonic and a scale that, together, create a framework of pitch relationships, both vertically and horizontally, within a musical composition [27].

However, key detection holds less significance in folk music, as vocal melodies are often transposed to accommodate a singer's vocal range, and instrumental tunes are frequently adjusted to suit the capabilities of specific instruments.

The following paragraphs will review existing literature on key detection, highlighting its relevance as a baseline for exploring mode detection in the analysis of folk music traditions.

Computational efforts in automatic key detection have supported various content-driven applications within musicology and music theory, notably facilitating large-scale empirical analysis [28,29]. This computational approach has found applications in music information retrieval tasks such as classification and retrieval [30–32], content analysis, and psychology, where it contributes to hierarchical representations of musical pitch [13,14,33]. The pursuit of automatic key detection extends to symbolic music encodings (e.g., MusicXML and MIDI) and audio recordings (Please refer to Zhu et al. [34] for a comprehensive review of audio key detection methods, which fall outside the scope of our work concerned with symbolic music encodings).

Supervised machine learning algorithms have also been extensively investigated and applied to key detection [17,18]. The utilization of labeled datasets and explicit guidance during the training phase allowed us to obtain accurate predictions in various musical contexts. Ref. [17] presents an approach using different supervised algorithms, such as KNN, SVM, and CNN. Furthermore, Ref. [18] tries to detect major and minor modes using an SVM algorithm. Finally, Ref. [9] tries to compare different Artificial Neural Networks (ANNs) to detect the minor and major modes.

However, the exploration of unsupervised learning approaches in key detection remains relatively limited in comparison. We can mention [35], which proposes an unsupervised approach to detect the major and minor modes based on Manhattan distance among the features of the data. Unsupervised methods have the potential to unveil underlying structures and relationships within musical data without explicit labeling, providing a more autonomous and adaptable approach. The scarcity of research in this area underscores the need for a more comprehensive exploration of unsupervised learning techniques in key detection.

3. Methods

The proposed system for the unsupervised classification of musical modes is organized as shown in Figure 1.

One of the goals of the present work is to check how the method of data preparation can influence the performance of algorithms in the classification procedure. In relation to these goals, an interesting comparison is to check how embedding approaches can capture such a high-level feature as the mode of a song compared to a more manual approach based on a selected feature that musicologists recommend: the pitch profiles.

Both paradigms belong to the preprocessing step. Therefore, there are two different flows in this stage, which merge in the second step of classification, where different unsupervised learning classifiers are applied to finally extract the classification results for both sets of preprocessed data.

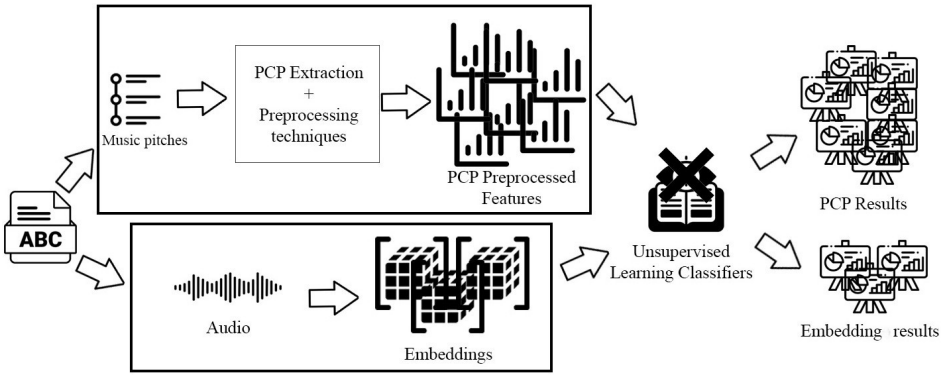

Figure 1. Proposed system workflow.

The workflow started from the initial dataset. This dataset consists of traditional Irish music in ABC notation, a text-based format for encoding musical notes and structure. These scores contain not only the digitalized notes but also metadata information about genre, key and mode. In particular, we used the preprocessed and revised version provided by Sturm et al. [36], which focuses primarily on monophonic songs and includes ABC fields that capture musical surface details and formal annotations, including mode. The authors reviewed the dataset fields twice to correct for any misclassified annotations. Additionally, all pieces were transposed to the tonic C. In total, the dataset comprises 23,636 folk songs, distributed among modes as follows: Ionian (15,861, 67.1%), Dorian (2971, 12.6%), Mixolydian (1620, 6.9%), and Aeolian (3184, 13.5%).

Once the dataset was prepared, the flow was divided. On the one hand, we extracted the Pitch Class Profiles (PCPs) through the music pitches provided by the ABC scores. PCPs represent the distribution of pitches across a 12-semitone octave. They are especially suitable for capturing harmonic information and have been widely used in tasks requiring mode or key identification [37]. We computed both the weighted and simple PCPs. In the weighted PCP, the pitch counts are adjusted by their temporal prominence within each

piece, whereas in the simple PCP, all pitches are treated equally regardless of duration or placement in the tune.

The simple PCP is computed as follows:

$$\textit{Simple PCP histogram}[c] = \sum_{i=1}^{N} \delta((p_i \mod 12) = c), \quad c \in \{0, 1, \ldots, 11\} \tag{1}$$

where p_i represents the MIDI pitch of the i-th note.

In the weighted PCP, durations (d_i) and beat strengths (b_i) adjust the contribution of each note:

$$\textit{Weighted PCP histogram}[c] = \sum_{i=1}^{N} d_i \cdot b_i \cdot \delta((p_i \mod 12) = c) \tag{2}$$

where d_i assigns a value of 1 to quarter notes, 0.5 to eighth notes, and so on, while b_i follows a hierarchy where strong beats are weighted more heavily than weak beats.

In summary, we applied several preprocessing techniques to the PCPs to enhance the representation and prepare the data for classification. These preprocessing techniques included the following:

- **Weighted and Simple PCPs.** As mentioned, both weighted and unweighted (simple) PCPs were computed, providing distinct representations of pitch prominence.
- **Dimensionality Reduction.** To reduce the complexity of PCP vectors, we applied Uniform Manifold Approximation and Projection (UMAP) [38] and Locally Linear Embedding (LLE) [39], two widely used nonlinear dimensionality reduction techniques. Each method was applied to both simple and weighted PCPs to examine their effects on classification accuracy and clustering performance. In particular, we selected the umap-learn implementation for UMAP reduction [40] and scikit-learn for LLE reduction [41].
- **Binary PCPs.** For further simplification, we generated binary PCPs from the simple PCP representations. These binary PCP vectors consist of 12 elements, each assigned a value of 1 if a pitch class appears in the piece or 0 if it does not. This binary approach provides a highly simplified representation that may highlight certain modal characteristics while ignoring specific pitch counts.

On the other hand, we obtained embeddings from a previous study on genre classification [19], which extracted features directly from ABC notation. In this work, each musical piece was converted into WAV audio format before applying different embedding models. The models used were JukeMIR, Mule, and MERT [42–44], each providing unique representations of audio.

- **JukeMIR.** This approach is currently one of the most prominent in the field, initially introduced as part of the Jukebox system, which generates music [43]. Jukebox utilizes the latest advancements in Natural Language Processing (NLP) by transforming raw audio signals into discrete sequences, which are then processed directly by NLP models, specifically Transformers. This approach enables Jukebox to train a Transformer-based autoregressive generative model using encoded audio data from an extensive dataset of one million songs. This model has proven to be effective in music generation. However, there is also another work, developed by Castellon et al. [45], that demonstrates that meaningful representations can be extracted from Jukebox by utilizing the activations of layer 36. This method generates a 4800-dimensional embedding that effectively represents each song. This one is the implementation used in this work to extract embeddings from the pieces.

- **Mule.** McCallum [42] recently developed a model that features various types of embeddings using both supervised and unsupervised approaches. In the supervised learning setup, large sets of log-mel spectrograms are used, sourced from both music and general audio domains. For unsupervised learning, SimCLR loss is applied to the same datasets, resulting in a more compact embedding (1728 dimensions) that requires fewer computational resources. In this work, we used the *sxmp-mule* Python package (version 1.1.2), which implements this algorithm. The default configuration was used.
- **MERT.** The MERT model [44] is designed to capture acoustic patterns inherent in musical audio signals. It employs a deep learning architecture with an initial unsupervised pretraining phase to detect salient spectrogram patterns, followed by a supervised fine-tuning phase. The model is given an unlabeled dataset, enabling it to align its learned representations with specific musical attributes, such as instrument classification. This convergence of unsupervised and supervised learning allows MERT to map raw audio spectrograms to a high-dimensional latent space, where musical information is compactly encoded. The network's final layer produces an N-dimensional vector representation that encapsulates the essence of the input audio, serving as an embedding that reflects the original audio waveform.In this work, we used the *transformers* Python package, particularly the *m-a-p/MERT-v1-330M* pretrained model of the *AutoModel class.*

After this large preprocessing step, we obtained all the representations. In summary, all of these combinations produced the following representations:

- Simple PCP, Weighted PCP, and Binary PCP, which have 12 dimensions;
- LLE reduction algorithm applied to Simple PCP, producing a 2-dimensional vector;
- LLE reduction algorithm applied to Weighted PCP, which produced a 2-dimensional vector;
- UMAP reduction algorithm applied to Simple PCP, which produced a 2-dimensional vector;
- UMAP reduction algorithm applied to Weighted PCP, which produced a 2-dimensional vector;
- The three embedding representations—JukeMIR, Mule and MERT—have 4800, 1728 and 1024 dimensions, respectively.

Then, we input the obtained representations into various classifiers for unsupervised mode classification. The classifiers were evaluated for both the simple and weighted PCP representations and the embeddings. Based on clustering metrics, we evaluated the effectiveness of each combination of preprocessing and classification to determine the optimal pipeline for unsupervised mode identification in traditional Irish music.

3.1. Unsupervised Learning Algorithms

This study employed five unsupervised learning algorithms to classify musical modes: Agglomerative Clustering, K-Means, DBSCAN, Mean Shift, and Self-Organizing Maps (SOM). These algorithms represent a range of classical clustering techniques [46–48]. Each algorithm was configured with specific parameters to improve its performance and adapt in to the characteristics of the data set.

- **K-Means** [49]. For K-Means, it is necessary to specify the number of clusters to be identified. This value was set to 4, corresponding to the four diatonic modes in the dataset: Ionian, Aeolian, Mixolydian, and Dorian. The scikit-learn implementation of this algorithm was used in this study [41].
- **Agglomerative Clustering** [50]. Agglomerative Clustering also requires the user to specify the number of clusters to be identified. As with the K-Means algorithm, the number of clusters was set to 4. The scikit-learn implementation of this algorithm was used [41].

- **DBSCAN (Density-Based Spatial Clustering of Applications with Noise)** [51]. DBSCAN relies on two key parameters: the minimum number of samples required to form a cluster (`min_samples`) and the maximum distance between points within a cluster (`eps`). In this study, `min_samples` was fixed at 5, the default value, as it offers a balance between identifying meaningful clusters and mitigating noise. The `eps` parameter was determined dynamically based on the characteristics of the input data. Specifically, a function was used to estimate the optimal `eps` by identifying the point at which the data distribution shifted significantly, ensuring well-separated clusters while minimizing noise. For instance, `eps` was set to 0.3 for PCP representations reduced with UMAP and 1.0 for Binary PCP representations. The scikit-learn implementation of this algorithm was used in this study [41].
- **Mean Shift** [52]. Mean Shift clustering requires the `bandwidth` parameter, which controls the search space size to identify clusters. In this study, the algorithm implementation automatically determined the optimal `bandwidth` based on the data distribution, eliminating the need for manual tuning. The scikit-learn implementation of this algorithm was used in this work [41].
- **Self-Organizing Maps (SOM)** [53]. The key parameter in SOM is the number of centroids, which defines the number of clusters to be identified. As with Agglomerative Clustering and K-Means, the number of centroids was set to 4, reflecting the four modes in the dataset. The MiniSom implementation of this algorithm was used [54].

These algorithms were chosen to explore a diverse range of clustering strategies, from centroid-based methods like K-Means to density-based techniques such as DBSCAN and Mean Shift, as well as hierarchical (Agglomerative Clustering) and neural network-based approaches (SOM).

3.2. Evaluation Metrics

In this work, we have computed several metrics to evaluate the quality of the clustering outcomes. These metrics provide insights into how well the clustering algorithms assign data points to their respective modes. Specifically, we calculated purity, which measures the alignment of clusters with the ground truth labels, as well as Normalized Mutual Information (NMI) and the Adjusted Rand Index (ARI), which assess the coherence and separation of the clusters in relation to the true labels.

- Purity measures how well each cluster is dominated by a single musical mode. A higher purity score indicates that most samples within a cluster belong to the same mode. We report the highest purity score for each mode.
- Normalized Mutual Information (NMI) [55] evaluates the agreement between the clustering results and the ground truth labels, accounting for the entropy of both. Unlike purity, NMI balances cluster homogeneity and completeness, making it robust to variations in cluster size. A higher NMI score indicates a better alignment between the clustering structure and the true mode labels.
- The Adjusted Rand Index (ARI) [56] assesses clustering quality by comparing the similarity between predicted and true labels, adjusting for random chance. The ARI rewards both correct cluster assignments and correct separations while penalizing incorrect merges or splits. Higher ARI values suggest better clustering performance.

4. Results and Discussion

This section presents the results and discussion of our classification problem. The goals of the experiments were threefold. Firstly, we aimed to demonstrate which PCP-based or embedding-based approach performs best according to our classification problem. Secondly, we aimed to check whether a featured-based approach yields better classification

results than a more automatic approach, such as the embedding approach. Finally, we examined how the unsupervised learning approach behaved with musical data when classifying mode, paying attention to the variety of algorithms that can be used in this paradigm.

MinMax normalization was applied to all PCP-based features to standardize the data and enhance the robustness of the classification process against variations in feature ranges (except PCP Binary because the feature consists of vectors whose elements are one or zero). Additionally, SMOTE (the Synthetic Minority Over-sampling Technique) was used to balance the class distribution, particularly for modes with fewer instances, such as Mixolydian and Dorian. Some internal experiments were carried out without using this technique to decide whether to apply it. The results showed a considerable improvement of the results when using SMOTE. In particular, we used the implementation from the imbalanced-learn Python package [57].

4.1. PCP Configuration Experiments

Tables 1 and 2 present the performance of the selected unsupervised algorithms on various PCP representations. Each row corresponds to a specific musical mode (Dorian, major, minor, or Mixolydian), while each column represents a different PCP configuration, as detailed in Section 3. The values in the table indicate the evaluation metric of purity for different classifiers. Furthermore, Tables 3 and 4 show the evaluation of Normalized Mutual Information (NMI) and the Adjusted Rand Index (ARI). The best results are highlighted in bold.

Table 1. Performance results of different classifiers using different types of PCP: Simple (Simple), Weighted (Weight), and Binary (Binary). The number in parentheses indicates the dominant cluster for each mode.

		Simple PCP	Weighted PCP	Binary PCP
K-Means	Dor	0.8826 (C1)	0.7737 (C1)	**0.7542 (C2)**
	Maj	0.9744 (C2)	0.7635 (C3)	**0.9798 (C3)**
	Min	0.6633 (C1)	0.7758 (C1)	**0.8066 (C4)**
	Mix	0.4600 (C1)	0.6678 (C1)	**0.6600 (C1)**
Agglom.	Dor	0.6329 (C2)	0.7062 (C4)	**0.6919 (C1)**
	Maj	0.6246 (C3)	0.8241 (C3)	**0.9152 (C3)**
	Min	0.7597 (C2)	0.6710 (C4)	**0.7280 (C4)**
	Mix	0.5189 (C4)	0.4798 (C3)	**0.9033 (C2)**
DBSCAN	Dor	0.9999 (C1)	0.99 (C1)	1.0 (C1)
	Maj	0.9992 (C1)	0.99 (C1)	0.9997 (C1)
	Min	0.9997 (C1)	0.99 (C1)	1.0 (C)1
	Mix	1.0 (C1)	0.99 (C1)	0.9956 (C1)
Mean Shift	Dor	0.9258 (C1)	0.9136 (C1)	0.9948 (C1)
	Maj	0.9244 (C1)	0.8813 (C1)	0.9992 (C1)
	Min	0.8785 (C1)	0.9018 (C1)	0.9664 (C1)
	Mix	0.9350 (C1)	0.9160 (C1)	0.9991 (C1)

Table 1. *Cont.*

		Simple PCP	Weighted PCP	Binary PCP
SOM	Dor	1.0 (C2)	1.0 (C4)	0.9707 (C1)
	Maj	1.0 (C2)	1.0 (C4)	0.9943 (C4)
	Min	1.0 (C2)	1.0 (C4)	0.9983 (C1)
	Mix	1.0 (C2)	1.0 (C4)	0.8259 (C4)

Table 2. Performance results of different classifiers using simple and weighted PCPs combined with LLE and UMAP reduction algorithms. The number in parentheses indicates the dominant cluster for each mode.

		Simple+LLE	Weighted+LLE	Simple+UMAP	Weighted+UMAP
K-Means	Dor	0.5634 (C2)	0.7437 (C1)	0.5820 (C2)	0.5201 (C1)
	Maj	0.6610 (C3)	0.5366 (C3)	0.5520 (C1)	0.5031 (C2)
	Min	0.6737 (C1)	0.7017 (C1)	0.9225 (C4)	0.7469 (C3)
	Mix	0.7767 (C2)	0.7869 (C1)	0.8220 (C2)	0.6238 (C1)
Agglom.	Dor	0.5282 (C3)	0.7083 (C3)	0.7329 (C3)	0.5244 (C2)
	Maj	0.7726 (C1)	0.6174 (C3)	0.5506 (C2)	0.6771 (C1)
	Min	0.6668 (C3)	0.6701 (C3)	0.9717 (C3)	0.8077 (C2)
	Mix	0.5847 (C4)	0.7800 (C3)	0.7178 (C1)	0.6240 (C4)
DBSCAN	Dor	1.0 (C1)	1.0 (C1)	0.9963 (C1)	1.0 (C1)
	Maj	1.0 (C1)	1.0 (C1)	0.9908 (C2)	1.0 (C1)
	Min	1.0 (C1)	1.0 (C1)	0.9989 (C1)	1.0 (C1)
	Mix	1.0 (C1)	1.0 (C1)	0.8270 (C1)	1.0 (C1)
Mean Shift	Dor	0.9953 (C1)	0.9685 (C1)	0.9936 (C1)	0.9711 (C1)
	Maj	0.9767 (C1)	0.9484 (C1)	0.9508 (C2)	0.9966 (C1)
	Min	0.9460 (C1)	0.9617 (C1)	0.9989 (C1)	0.9784 (C1)
	Mix	0.9937 (C1)	0.9764 (C1)	0.8270 (C1)	0.5347 (C1)
SOM	Dor	1.0 (C3)	1.0 (C2)	**0.9766 (C2)**	1.0 (C1)
	Maj	1.0 (C3)	1.0 (C2)	**0.4459 (C3)**	1.0 (C1)
	Min	1.0 (C3)	1.0 (C2)	**0.6453 (C1)**	0.9784 (C1)
	Mix	1.0 (C3)	1.0 (C2)	**0.7107 (C4)**	0.5216 (C1)

In the tables of results, each cell reports the purity score along with the dominant cluster number for each mode, providing insight into how well the algorithm distinguishes between musical modes. Furthermore, NMI and ARI scores offer a broader evaluation of clustering performance beyond purity alone. Although high purity indicates that individual clusters are dominated by a single mode, NMI and ARI help to assess whether the modes are meaningfully differentiated rather than arbitrarily grouped. Cases where multiple modes are assigned to the same cluster despite high purity suggest that the algorithm has failed to achieve proper separation. Thus, a comprehensive evaluation must consider all three metrics to ensure that the four musical modes are not only internally consistent within clusters but also well differentiated from each other.

Table 3. NMI and ARI scores for different clustering methods across PCP representations.

		Simple PCP	Weighted PCP	Binary PCP
K-Means	NMI	0.2535	0.2535	**0.5807**
	ARI	0.65	0.1657	**0.5626**
Agglom.	NMI	0.2348	0.2348	**0.5976**
	ARI	0.52	0.1909	**0.5756**
DBSCAN	NMI	0.0	0.0	0.0
	ARI	0.98	0.0	0.0
Mean Shift	NMI	0.0	0.0	0.0
	ARI	0.75	0.0	0.0
SOM	NMI	0.0	0.0	0.5105
	ARI	1.00	0.0	0.4065

Table 4. NMI and ARI scores for different classifiers using simple and weighted PCPs combined with LLE and UMAP reduction algorithms.

		Simple+LLE	Weighted+LLE	Simple+UMAP	Weighted+UMAP
K-Means	NMI	0.3333	0.1838	**0.5322**	0.3868
	ARI	0.2442	0.0793	**0.4339**	0.2942
Agglom.	NMI	0.3453	0.1216	0.5081	0.4048
	ARI	0.2752	0.0342	0.4110	0.3270
DBSCAN	NMI	0.0	0.0	0.4702	0.0
	ARI	0.0	0.0	0.3085	0.0
Mean Shift	NMI	0.0	0.0	0.4625	0.0
	ARI	0.0	0.0	0.2986	0.0
SOM	NMI	0.0	0.0	**0.5165**	0.3647
	ARI	0.0	0.0	**0.4672**	0.2387

As we can see, the experiments reveal distinct performance differences among the various combinations of PCP representations and clustering algorithms. Binary PCP, particularly when used with the Agglomerative and K-Means algorithms, achieved the most promising results, successfully identifying the four target clusters while maintaining over 60% purity across all modes. This was in contrast to other representations, such as the UMAP projection of Simple PCP (Simple+UMAP), which also succeeded in identifying the four clusters but showed diminished performance, particularly with the Dorian mode, for which it achieved only 47% purity.

The Binary PCP representation appears to enhance mode separability, likely due to its simplified representation, which focuses solely on the presence or absence of pitch classes. This binary simplification effectively removes extraneous details that may otherwise obscure modal patterns, making it particularly well suited for clustering algorithms such as Agglomerative and K-Means, which rely on clear, distinguishable groupings. Consequently, these two methods consistently performed better in clustering the modes with the Binary PCP than with other PCP representations.

This trend is further supported by the Normalized Mutual Information (NMI) and Adjusted Rand Index (ARI) results. The highest scores were obtained with Binary PCP and Agglomerative Clustering (NMI = 0.5976, ARI = 0.5756), followed closely by Binary PCP and K-Means (NMI = 0.5807, ARI = 0.5626). In contrast, while Simple PCP+UMAP combined with SOM managed to separate the four clusters, its performance was notably lower (NMI = 0.5165, ARI = 0.4672), aligning with its lower purity results. These findings

reinforce that the best-performing configurations not only achieved higher purity but also exhibited superior cluster alignment, as reflected in their NMI and ARI scores.

4.2. Comparing Embedding and PCP Approaches

Tables 5 and 6 show a comparison between the PCP Binary representation and the embeddings models (JukeMIR, Mule, and MERT). The best outcomes are highlighted in bold. These results clearly demonstrate the superiority of the Binary PCP representation in mode classification. While the Binary PCP achieves consistently high performance across all clustering algorithms, the embedding models struggle to effectively distinguish between the different modes. This limitation is evident in the clustering results, where a significant proportion of the data are assigned to the same cluster, suggesting that the embeddings do not capture the subtle tonal differences required for mode identification.

Table 5. Comparison of clustering purity for different embedding model representations and PCP Binary across modes. The number in parentheses indicates the dominant cluster for each mode.

		JukeMIR	Mule	MERT	PCP Binary
K-Means	Dor	0.4794 (C4)	0.7371 (C1)	0.3691 (C2)	**0.7542 (C2)**
	Maj	0.4791 (C4)	0.7365 (C1)	0.3921 (C2)	**0.9798 (C3)**
	Min	0.4935 (C4)	0.6923 (C1)	0.3893 (C2)	**0.8066 (C4)**
	Mix	0.4678 (C4)	0.7414 (C1)	0.4326 (C2)	**0.6600 (C1)**
Agglom.	Dor	0.4185 (C2)	0.6541 (C3)	0.4231 (C1)	**0.6919 (C1)**
	Maj	0.4097 (C2)	0.6672 (C3)	0.4120 (C1)	**0.9152 (C3)**
	Min	0.4207 (C2)	0.6234 (C3)	0.4032 (C1)	**0.7280 (C4)**
	Mix	0.4145 (C1)	0.6740 (C3)	0.4406 (C1)	**0.9033 (C2)**
DBSCAN	Dor	0.9229 (C1)	1.0 (C1)	0.9475 (C1)	1.0 (C1)
	Maj	0.8970 (C1)	1.0 (C1)	0.9381 (C1)	0.9997 (C1)
	Min	0.9199 (C1)	1.0 (C1)	0.8886 (C1)	1.0 (C1)
	Mix	0.8830 (C1)	1.0 (C1)	0.9575 (C1)	0.9956 (C1)
Mean Shift	Dor	0.9323 (C1)	0.8392 (C1)	0.9417 (C1)	0.9948 (C1)
	Maj	0.9208 (C1)	0.8364 (C1)	0.9273 (C1)	0.9992 (C1)
	Min	0.9325 (C1)	0.8108 (C1)	0.9435 (C1)	0.9664 (C1)
	Mix	0.9220 (C1)	0.8229 (C1)	0.9326 (C1)	0.9991 (C1)
SOM	Dor	1.0 (C4)	1.0 (C3)	1.0 (C4)	0.99 (C2)
	Maj	1.0 (C4)	1.0 (C3)	1.0 (C4)	0.99 (C2)
	Min	1.0 (C4)	1.0 (C3)	1.0 (C4)	0.99 (C2)
	Mix	1.0 (C4)	1.0 (C3)	1.0 (C4)	0.99 (C2)

The embedding-based approach introduces additional complexity by representing each tune within a high-dimensional feature space (e.g., 4800 dimensions for JukeMIR), which may obscure the fundamental characteristics that differentiate the modes. For instance, the embeddings tend to overfit certain acoustic patterns in the data, leading to reduced generalization across clusters. Conversely, the Binary PCP's simplified representation emphasizes the presence or absence of pitches, allowing the clustering algorithms to focus on the distinctive pitch-class distributions of each mode. This is particularly critical for distinguishing between modes that differ by only one or two notes, such as Ionian and Mixolydian or Dorian and Aeolian.

Table 6. Comparison of clustering NMI and ARI values for different embedding model representations and PCP Binary across modes.

		JukeMIR	Mule	MERT	PCP Binary
K-Means	NMI	0.0062	0.0005	0.0055	**0.5807**
	ARI	0.0019	0.0091	0.0095	**0.5626**
Agglom.	NMI	0.0061	0.0005	0.0091	**0.5976**
	ARI	0.0004	0.0065	0.0025	**0.5756**
DBSCAN	NMI	0.0	0.0	0.0	0.0
	ARI	0.0	0.0	0.0	0.0
Mean Shift	NMI	0.0	0.0	0.0	0.0
	ARI	0.0	0.0	0.0	0.0
SOM	NMI	0.0	0.0	0.0	0.5105
	ARI	0.0	0.0	0.0	0.4065

Overall, the results suggest that embedding models, while powerful in other music-related tasks, may not be well suited for mode classification in this specific context. Their inability to consistently detect the unique characteristics of diatonic modes underscores the advantages of using tailored, domain-specific features such as PCP Binary for unsupervised classification.

4.3. Unsupervised Learning Algorithms

As shown in all the codification outcomes, clustering algorithms that do not require a predefined number of clusters, such as DBSCAN, SOM, and Mean Shift, encountered significant challenges in this study. Unlike K-Means and Agglomerative Clustering, which explicitly define the number of clusters, these methods dynamically determine cluster boundaries based on the data distribution. This approach often resulted in an inaccurate number of clusters, created by grouping either too many or too few samples.

In the case of DBSCAN, the algorithm struggled to identify the four distinct musical modes due to its sensitivity to the `eps` parameter, which defines the maximum distance between points in a cluster. Depending on the density of the input data, DBSCAN sometimes produced fewer clusters, merging modes into a single group, or fragmented clusters, leading to over-segmentation. Similarly, Mean Shift faced difficulties in automatically selecting an optimal bandwidth, resulting in inconsistent cluster formation between different feature sets. SOM also failed to cluster effectively in most cases, classifying all data into a single cluster regardless of the predefined centroids. Notably, SOM only achieved some separation when the UMAP reduction was applied to the dataset.

These limitations highlight the importance of algorithms such as K-Means and Agglomerative Clustering when the number of clusters is known a priori. These algorithms consistently aligned with the dataset's structure by specifying the number of clusters (in this case, four), accurately representing the distinct modes. Consequently, for tasks requiring the classification of predefined categories, algorithms with configurable cluster counts are more reliable and effective.

5. Conclusions

In this study, our main objective was to determine the best preprocessing techniques in the task of mode detection in folk music, using Irish folk music as an example and focusing the work on unsupervised learning paradigm. To reach this objective, multiple experiments were performed to test different methods, such as different PCPs, embeddings, and reduction algorithms. In summary, the results highlight that the Binary PCP representation is the most effective feature for unsupervised mode classification of Irish folk music. Its

simplicity and focus on pitch-class presence enable clustering algorithms to accurately identify the four diatonic modes, outperforming both the more complex embeddings and the other PCP variations. Among the clustering algorithms, K-Means and Agglomerative Clustering consistently deliver strong performance, achieving high accuracy in identifying clusters and correctly classifying modes. In contrast, DBSCAN and Mean Shift struggle due to their sensitivity to parameter selection and the lack of predefined clusters, leading to inconsistent results. Similarly, the embeddings, while offering high-dimensional representations, introduce unnecessary complexity and fail to capture the subtle tonal differences that distinguish the modes. These findings emphasize the importance of selecting appropriate feature representations and algorithms tailored to the specific task of diatonic mode classification.

Finally, for future work, this study opens opportunities to explore unsupervised mode classification in other musical traditions and cultures, particularly those with modal systems distinct from Western diatonic modes. Extending the analysis to non-Western music could reveal how these techniques generalize to different modal frameworks and highlight potential adaptations required for culturally specific features. Additionally, comparing the performance of unsupervised learning and supervised approaches could provide deeper insights into the trade-offs between autonomy and accuracy. By incorporating labeled datasets, supervised methods could be used as a benchmark to evaluate the efficacy of unsupervised pipelines, offering a more comprehensive understanding of their strengths and limitations in mode classification tasks.

Author Contributions: Conceptualization, J.J.N.-C., D.M.J.-B. and M.N.-C.; methodology, J.J.N.-C., D.M.J.-B. and M.N.-C.; software, J.J.N.-C.; validation, J.J.N.-C., D.M.J.-B. and M.N.-C.; formal analysis, J.J.N.-C., D.M.J.-B. and M.N.-C.; investigation, J.J.N.-C.; resources, J.J.N.-C., D.M.J.-B. and M.N.-C.; data curation, J.J.N.-C.; writing—original draft preparation, J.J.N.-C.; writing—review and editing, D.M.J.-B. and M.N.-C.; visualization, J.J.N.-C.; supervision, D.M.J.-B. and M.N.-C.; project administration, M.N.-C.; funding acquisition, M.N.-C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was partially funded by the project "EA-DIGIFOLK: An European and Ibero-American approach for the digital collection, analysis and dissemination of folk music" (101086338) under the program Marie Skłodowska-Curie Actions Staff Exchanges (HORIZON-MSCA-2021-SE-01-01).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original data presented in the study are openly available from The Session at https://thesession.org/ (accessed on 25 February 2025).

Acknowledgments: We acknowledge the use of ChatGPT-4 for text editing and refinement support in the manuscript preparation.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Manzano, M. Aspectos metodológicos en la investigación etnomusicólogica. *Rev. Musicol.* **1997**, *20*, 991–1002.
2. Nettl, B. *The Study of Ethnomusicology: Twenty-Nine Issues and Concepts*; University of Illinois Press: Champaign, IL, USA, 1983; Number 39.
3. Misto, R. Therapeutic Musical Scales: Theory and Practice. *OBM Integr. Complement. Med.* **2021**, *6*, 019. [CrossRef]
4. Egan, P. Insider or outsider? Exploring some digital challenges in ethnomusicology. *Interdiscip. Sci. Rev.* **2021**, *46*, 477–500.
5. Gómez, E.; Herrera, P.; Gómez-Martin, F. Computational ethnomusicology: Perspectives and challenges. *J. New Music. Res.* **2013**, *42*, 111–112. [CrossRef]

6. Navarro-Cáceres, J.J.; Carvalho, N.; Bernardes, G.; Jiménez-Bravo, D.M.; Navarro-Cáceres, M. Exploring Mode Identification in Irish Folk Music with Unsupervised Machine Learning and Template-Based Techniques. In *Mathematics and Computation in Music*; Noll, T., Montiel, M., Gómez, F., Hamido, O.C., Besada, J.L., Martins, J.O., Eds.; Springer: Cham, Switzerland, 2024; pp. 412–420.
7. Bertin-Mahieux, T.; Ellis, D.P.; Whitman, B.; Lamere, P. The Million Song Dataset. In Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR 2011), Miami, FL, USA, 24–28 October 2011.
8. Gajjar, P.; Shah, P.; Sanghvi, H. E-mixup and siamese networks for musical key estimation. In Proceedings of the International Conference on Ubiquitous Computing and Intelligent Information Systems 2021, Tamil Nadu, India, 16–17 April 2021; Springer: Singapore, 2021; pp. 343–350.
9. Garg, M.; Gajjar, P.; Shah, P.; Shukla, M.; Acharya, B.; Gerogiannis, V.C.; Kanavos, A. Comparative Analysis of Deep Learning Architectures and Vision Transformers for Musical Key Estimation. *Information* **2023**, *14*, 527. [CrossRef]
10. Tzanetakis, G.; Cook, P. Musical genre classification of audio signals. *IEEE Trans. Speech Audio Process.* **2002**, *10*, 293–302. [CrossRef]
11. Eerola, T.; Toiviainen, P. Suomen Kansan eSävelmät. Finnish Folk Song Database. [11.3. 2004]. 2004. Available online: http://www.jyu.fi/musica/sks/ (accessed on 15 December 2024).
12. Session, T. Traditional Irish Nusic on The Session. 2021. Available online: https://thesession.org/ (accessed on 15 December 2024).
13. Krumhansl, C.L.; Kessler, E.J. Tracing the dynamic changes in perceived tonal organization in a spatial representation of musical keys. *Psychol. Rev.* **1982**, *89*, 334. [CrossRef]
14. Krumhansl, C.L. *Cognitive Foundations of Musical Pitch*; Oxford University Press: Oxford, UK, 2001; Volume 17.
15. Temperley, D. What's key for key? The Krumhansl-Schmuckler key-finding algorithm reconsidered. *Music. Percept.* **1999**, *17*, 65–100. [CrossRef]
16. Jansson, A.; Weyde, T. MIREX 2012: Key recognition with zweiklang profiles. In *Music Information Retrieval Evaluation eXchange (MIREX)*; Citeseer: University Park, PA, USA, 2012.
17. George, A.; Mary, X.A.; George, S.T. Development of an intelligent model for musical key estimation using machine learning techniques. *Multimed. Tools Appl.* **2022**, *81*, 19945–19964. [CrossRef]
18. Mahieu, R. Detecting Musical Key with Supervised Learning. 2017. Available online: https://api.semanticscholar.org/CorpusID:38498183 (accessed on 15 December 2024).
19. Jimenez-Bravo, D.M.; Lozano-Murciego, A.; Navarro-Caceres, J.J.; Navarro-Caceres, M.; Harkin, T. Identifying Irish Traditional Music Genres Using Latent Audio Representations. *IEEE Access* **2024**, *12*, 92536–92548. [CrossRef]
20. Cai, X.; Zhang, H. Fisher Discriminative Embedding Low-Rank Sparse Representation for Music Genre Classification. *Circuits Syst. Signal Process.* **2024**, *43*, 5139–5168. [CrossRef]
21. Liu, X.; Yang, Z.; Cheng, J. Music recommendation algorithms based on knowledge graph and multi-task feature learning. *Sci. Rep.* **2024**, *14*, 2055. [CrossRef]
22. Tabaza, A.; Quishawi, O.; Yaghi, A.; Qawasmeh, O. Binding Text, Images, Graphs, and Audio for Music Representation Learning. In Proceedings of the AICCONF'24: Cognitive Models and Artificial Intelligence Conference, İstanbul, Turkiye, 25–26 May 2024; pp. 139–146. [CrossRef]
23. Marijić, A.; Bagić Babac, M. Predicting song genre with deep learning. *Glob. Knowl. Mem. Commun.* **2025**, *74*, 93–110. [CrossRef]
24. Huang, Q.; Jansen, A.; Lee, J.; Ganti, R.; Li, J.Y.; Ellis, D.P.W. MuLan: A Joint Embedding of Music Audio and Natural Language. *arXiv* **2022**, arXiv:2208.12415.
25. Doh, S.; Lee, J.; Park, T.H.; Nam, J. Musical Word Embedding: Bridging the Gap between Listening Contexts and Music. *arXiv* **2020**, arXiv:2008.01190.
26. Dokania, S.; Singh, V. Graph Representation learning for Audio & Music genre Classification. *arXiv* **2019**, arXiv:1910.11117.
27. Milne, A. *A Computational Model of the Cognition of Tonality*; Open University (United Kingdom): Milton Keynes, UK, 2013.
28. Quinn, I.; White, C.W. Corpus-derived key profiles are not transpositionally equivalent. *Music. Percept. Interdiscip. J.* **2017**, *34*, 531–540. [CrossRef]
29. Quinn, I. Are pitch-class profiles really "Key for Key"? *Z. Der Ges. FÜR Musik. [J.-Ger.-Speak. Soc. Music. Theory]* **2010**, *7*, 151–163. [CrossRef] [PubMed]
30. Bernardes, G.; Davies, M.E.; Guedes, C. Automatic musical key estimation with adaptive mode bias. In Proceedings of the 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), New Orleans, LA, USA, 5–9 March 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 316–320.
31. Chuan, C.H.; Chew, E. Polyphonic audio key finding using the spiral array CEG algorithm. In Proceedings of the 2005 IEEE International Conference on Multimedia and Expo, Amsterdam, The Netherlands, 6 July 2005; IEEE: Piscataway, NJ, USA, 2005; pp. 21–24.
32. Gómez, E.; Herrera, P. Estimating The Tonality Of Polyphonic Audio Files: Cognitive Versus Machine Learning Modelling Strategies. In Proceedings of the ISMIR 2004, Barcelona, Spain, 10–14 October 2004.

33. Huron, D.; Parncutt, R. An improved model of tonality perception incorporating pitch salience and echoic memory. *Psychomusicology J. Res. Music. Cogn.* **1993**, *12*, 154. [CrossRef]
34. Zhu, Y.; Kankanhalli, M.S. Precise pitch profile feature extraction from musical audio for key detection. *IEEE Trans. Multimed.* **2006**, *8*, 575–584.
35. Finley, M.; Razi, A. Musical key estimation with unsupervised pattern recognition. In Proceedings of the 2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 7–9 January 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 0401–0408.
36. Sturm, B.; Santos, J.F.; Korshunova, I. Folk music style modelling by recurrent neural networks with long short term memory units. In Proceedings of the 16th International Society for Music Information Retrieval Conference, Malaga, Spain, 26–30 October 2015.
37. Cabral, G.; Briot, J.P.; Pachet, F. Impact of distance in pitch class profile computation. In Proceedings of the Brazilian Symposium on Computer Music 2005, Belo Horizonte, Brazil, 3–5 October 2025; pp. 319–324.
38. Sainburg, T.; McInnes, L.; Gentner, T.Q. Parametric UMAP: Learning embeddings with deep neural networks for representation and semi-supervised learning. *arXiv* **2020**, arXiv:2009.12981.
39. Roweis, S.T.; Saul, L.K. Nonlinear Dimensionality Reduction by Locally Linear Embedding. *Science* **2000**, *290*, 2323–2326. [CrossRef]
40. McInnes, L.; Healy, J.; Melville, J. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv* **2018**, arXiv:1802.03426.
41. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
42. McCallum, M.C.; Korzeniowski, F.; Oramas, S.; Gouyon, F.; Ehmann, A.F. Supervised and Unsupervised Learning of Audio Representations for Music Understanding. *arXiv* **2022**, arXiv:2210.03799.
43. Dhariwal, P.; Jun, H.; Payne, C.; Kim, J.W.; Radford, A.; Sutskever, I. Jukebox: A Generative Model for Music. *arXiv* **2020**, arXiv:2005.00341.
44. Li, Y.; Yuan, R.; Zhang, G.; Ma, Y.; Chen, X.; Yin, H.; Xiao, C.; Lin, C.; Ragni, A.; Benetos, E.; et al. MERT: Acoustic Music Understanding Model with Large-Scale Self-supervised Training. *arXiv* **2023**, arXiv:2306.00107.
45. Castellon, R.; Donahue, C.; Liang, P. Codified audio language modeling learns useful representations for music information retrieval. *arXiv* **2021**, arXiv:2107.05677. [CrossRef]
46. Cao, L.; Zhao, Z.; Wang, D. Clustering algorithms. In *Target Recognition and Tracking for Millimeter Wave Radar in Intelligent Transportation*; Springer: Berlin/Heidelberg, Germany, 2023; pp. 97–122.
47. Venkatkumar, I.A.; Shardaben, S.J.K. Comparative study of data mining clustering algorithms. In Proceedings of the 2016 International Conference on Data Science and Engineering (ICDSE), Cochin, India, 23–25 August 2016; pp. 1–7. [CrossRef]
48. Carreira-Perpinán, M.A. A review of mean-shift algorithms for clustering. *arXiv* **2015**, arXiv:1503.00687.
49. Arthur, D.; Vassilvitskii, S. k-means++: The advantages of careful seeding. In Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA'07, New Orleans, LA, USA, 7–9 January 2007; pp. 1027–1035.
50. Ackermann, M.R.; Blömer, J.; Kuntze, D.; Sohler, C. Analysis of agglomerative clustering. *Algorithmica* **2014**, *69*, 184–215. [CrossRef]
51. Ester, M.; Kriegel, H.P.; Sander, J.; Xu, X. A density-based algorithm for discovering clusters in large spatial databases with noise. In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining KDD'96, Portland, OR, USA, 2–4 August 1996; AAAI Press: Washington, DC, USA, 1996; pp. 226–231.
52. Comaniciu, D.; Meer, P. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**, *24*, 603–619. [CrossRef]
53. Kohonen, T. The Self-Organizing Map. *Proc. IEEE* **1990**, *78*, 1464–1480. [CrossRef]
54. Vettigli, G. MiniSom: Minimalistic and NumPy-Based Implementation of the Self Organizing Map. 2018. Available online: https://github.com/JustGlowing/minisom (accessed on 15 December 2024).
55. Vinh, N.X.; Epps, J.; Bailey, J. Information theoretic measures for clusterings comparison: Is a correction for chance necessary? In Proceedings of the 26th Annual International Conference on Machine Learning, ICML'09, Montreal, QC, Canada, 14–18 June 2009; pp. 1073–1080. [CrossRef]
56. Steinley, D. Properties of the Hubert-Arabie Adjusted Rand Index. *Psychol. Methods* **2004**, *9*, 386–396. [CrossRef] [PubMed]
57. Lemaître, G.; Nogueira, F.; Aridas, C.K. Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning. *J. Mach. Learn. Res.* **2017**, *18*, 1–5.