Received December 12, 2020, accepted January 18, 2021, date of publication January 22, 2021, date of current version February 2, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3053864

Web-Based Music Genre Classification for Timeline Song Visualization and Analysis

JAIME RAMIREZ CASTILLO AND M. JULIA FLORES
Computing Systems Department, UCLM, 02071 Albacete, Spain

Corresponding author: Jaime Ramirez Castillo (jaime.ramirez@alu.uclm.es)

This work was supported in part by the Spanish Government [Agencia Española de Investigación (AEI)], in part by the Fondo Europeo de Desarrollo Regional (FEDER) funds, and in part by the Junta de Comunidades de Castilla-La Mancha (JCCM) under Grant PID2019-106758GB-C33/AEI/10.13039/501100011033 and Grant SBPLY/17/180501/000493.

ABSTRACT This paper presents a web application that retrieves songs from YouTube and classifies them into music genres. The tool explained in this study is based on models trained using the musical collection data from Audioset. For this purpose, we have used classifiers from distinct Machine Learning paradigms: Probabilistic Graphical Models (Naive Bayes), Feed-forward and Recurrent Neural Networks and Support Vector Machines (SVMs). All these models were trained in a multi-label classification scenario. Because genres may vary along a song's timeline, we perform classification in chunks of ten seconds. This capability is enabled by Audioset, which offers 10-second samples. The visualization output presents this temporal information in real time, synced with the music video being played, presenting classification results in stacked area charts, where scores for the top-10 labels obtained per chunk are shown. We briefly explain the theoretical and scientific basis of the problem and the proposed classifiers. Subsequently, we show how the application works in practice, using three distinct songs as cases of study, which are then analyzed and compared with online categorizations to discuss models performance and music genre classification challenges.

INDEX TERMS Classification algorithms, deep learning, machine learning, music information retrieval, probabilistic models, visualization, Web sites.

I. INTRODUCTION

Research in Music Information Retrieval (MIR) [1] comprises a broad range of topics including genre classification, recommendation, discovery and visualization. In short, this research line refers to knowledge discovery from music and involves its processing, study and analysis. When combined with Machine Learning techniques, we typically try to learn models able to emulate human abilities or tasks, which, if automated, can be helpful for the final user. Computational algorithms and models have even been applied for music generation and composition [2]–[4].

Music genre classification (MGC) is a discipline of the music annotation domain that has recently received attention from the MIR research community, especially since the seminal study of Tzanetakis and Cook [5]. The main objective in MGC is to classify a musical piece into one or more musical genres. As simple as it sounds, the field still presents challenges related to the lack of standardization and vague genre definitions. Public databases and ontologies do not usually agree on how each genre is defined. Moreover, human music perception, subject to opinions and personal experiences, makes this agreement even more difficult. For example, when a song includes swing rhythms, piano, trumpets and improvisation, we would probably define it as jazz music. However, if we introduce synthesizers in the same song, should the song be classified as electronic music as well? If we only consider acoustic characteristics, the answer is probably yes. But different listeners can perceive the piece from their own perspective. Whereas some might categorize the song as jazz, others might consider it electronic music or even a combination of both.

In an effort to provide a tool that gives more insights about how each genre is perceived, we have trained several classification models [6] and embedded them in a web application that allows the user to visualize how each model "senses" music in terms of music genre, at particular moments of a song. Note that experimentation details for each model are beyond the scope of this article and can be found in [6]. These models have been built using common machine learning techniques, namely, Support Vector Machines (SVM), Naive Bayes classifiers, Feed forward deep neural networks

The associate editor coordinating the review of this manuscript and approving it for publication was Liangxiu Han.

and Recurrent neural networks. Whereas Bayesian and SVM methods have historically delivered good results as general-purpose machine learning models, the results achieved with deep learning techniques in artificial perception (artificial vision, speech recognition, natural language processing, among others) have delivered remarkable results, approaching human-like accuracy [7]. By comparing deep learning with more traditional machine learning techniques, we also aim to compare its performance for music genre classification.

The rest of the paper is organized as follows. In section 2, we present the problem and the models used. Sections 3 and 4 give an overview of the state of the art, the dataset and the experimentation results that support the application. Section 5 describes how the application works, its inputs, outputs and data flow. Section 6 includes three use cases to present the different behaviors of the models. In Section 7, we discuss the results of the scenarios from the previous section. Finally, in section 8, we conclude the article and discuss our thoughts for the future.

II. MACHINE LEARNING FRAMEWORK

Machine Learning (ML) is an area of Computer Science that involves the application of Artificial Intelligence techniques to learn from data. In our case, we perform the task of supervised classification. Taking a set of songs as input, labeled by genre, we have learned different models. The songs are characterized by specific features and the labels will guide the learning process. In this case, one song can be labeled with multiple genres, and they are classified in excerpts, as we will explain later. So, the problem that we approach in this work is the annotation of music genres present in a music clip, with the purpose of comparing the performance of different machine learning models when applied to this specific problem. To this end, we use the Audioset repository and its music genre samples to train the following set of models.

A. DECISION TREES

Decision trees are an appealing option for classification. A decision tree is a collection of nodes, connected by branches, extending downwards from the root node to leaf nodes. Beginning at the root, attributes are tested at the decision nodes, with each possible outcome resulting in a branch. Decision trees can be trained with the classic ID3 algorithm and also with more complex solutions, such as C4.5 and C5.0. These algorithms use information gain measures to establish which variable is more informative with respect to the target or class, in an incremental and recursive process. ID3 generates too deep and complex trees, which tend to overfit, i.e. they are excessively faithful to training data and generalize poorly. More sophisticated algorithms perform pruning steps to overcome this issue. An example of a decision tree can be seen in figure 1.(a). In this particular case, it illustrates a very simple problem for deciding whether a bank credit is given to a client. For example, with the input case {Expenses=500, Owner=Yes and Income=Yes}, the decision will be *Yes*. Notice that in this particular case the Income value is irrelevant, as it is not tested. Decision trees have been used successfully in music classification [8], [9].

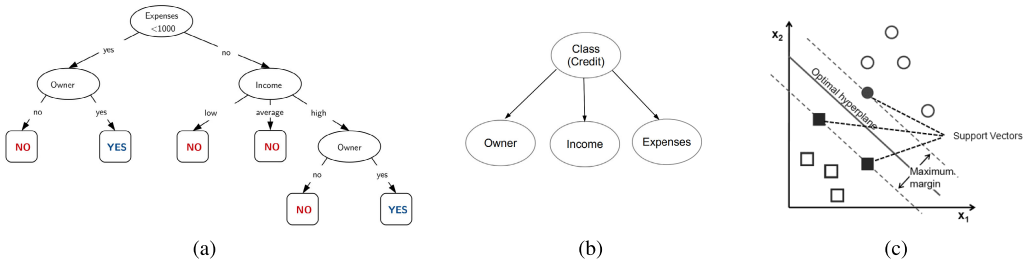
In our experiments, Decision Trees achieve a mean average precision (AP) of 0.060 and a ROC area under the curve (AUC) of 0.560.

B. PROBABILISTIC CLASSIFIERS: NAIVE BAYES

Naive Bayes (NB) classifiers belong to the family of probabilistic models. In particular, they are categorized within Bayesian Networks, which present two main components: (1) the graphical model representation – DAG (Directed Acyclic Graph) – with a set of nodes and edges, and (2) a joint probability distribution (JPD). Nodes represent the variables in the problem domain and edges refer to their direct relationships. Traditionally, they have also been called causal networks as these relationships could be read as $\boxed{\text{Cause}} \longrightarrow \boxed{\text{Effect}}$, but this is not a real requirement. The JPD will be encoded by the graphical model but in a simplified way, as these direct relationships provide a factorization, and enable the definition of (in)dependence criteria [10]. Assuming the variables are named X_i and $i \in \{1, \ldots, n\}$, we can express and compute the JPD as $P(\overline{X}) = P(X_i, \ldots, X_n) = \prod_{i=1}^{n} P(X_i|pa(X_i))$, where $pa(X_i)$ refers to the parents of node X_i, that is, those having an arrow pointing to it. This factorization can be guaranteed by the chain rule [10] and is possible due to the absence of directed cycles.

Bayesian network classifiers have been used in many different domains [11], but the most representative model is undoubtedly Naive Bayes. This type of classifier relies on the Bayes theorem and, given the class value, assumes independence between input features. This assumption is naive, in such a way that all input features are presumed to contribute equally and independently to the target class. NB can be seen as a special case of a Bayesian network, specially intended for classification. Whereas in typical Bayesian networks we would place the emphasis on accurately estimating the joint probability distribution, in a NB classifier we give priority to correctly estimating the target class. In Figure 1.(b), we can see a very simple model, where only three predictive variables are included. In this case, for classification, the model needs to have specified the values for the prior probability $P(Class)$ and $P(Income|Class)$, $P(Expenses|Class)$ and $P(Owner|Class)$. These probabilities will be estimated from the training data. When performing classification, given a specific configuration for Income, Expenses and Owner, the model will predict a probability distribution for the class, in this case *Yes*/*No* and the one with the higher probability will be chosen as the label.

Although Bayesian methods have been traditionally used in text analysis topics, they also appear in studies related to music analysis and retrieval [12], [13], music recommendation [14], [15], and music emotional perception [16].



(a) (b) (c)

FIGURE 1. Illustrations for toy examples of (a) Decision Tree, (b) Naive Bayes and (c) SVM.

In our experiments, the NB classifier achieves a mean average precision (AP) of 0.196 and a ROC area under the curve (AUC) of 0.830.

C. SUPPORT VECTOR MACHINES (SVMs)

Support Vector Machines generate hyperplanes that work as decision boundaries in high dimensional spaces to find optimal divisions between points of different classes. In short, they try to determine the best and broadest decision boundary between different classes [17].

Figure 1.(c) illustrates these models. In this simple example, there are only two predictive variables: x_1 and x_2. The class variable differs depending on the shape circle/square. The SVM is just a straight line in this illustration to show the main idea. In real SVMs, the decision boundary does not have to be a line. They are referred to as a hyperplane because you can find the decision boundary with any number of features. With the so-called large margin classification (see 1.(c) for a visual representation), these hyperplanes maximize the margin between elements from different classes. Those instances that mark the separation margin at both sides of the hyperplane are the support vectors. When data is not linearly separable, more flexible approaches can be used. For example, with soft margin classification, the model tries to balance the separation margin and the number of instances in the wrong side of the hyperplane. More advanced solutions can entail the use of polynomial features or kernel functions.

Examples of SVMs applied to music annotation can be found in [18], [19].

The linear SVM from our experiments gives a mean AP of 0.126 and an AUC of 0.596.

D. FULLY CONNECTED NEURAL NETWORKS (FCNNs)

FCNNs are part of a larger, popular area in ML and artificial intelligence called Deep Learning. Deep learning covers different ML techniques based on deep artificial neural networks (ANNs). These networks are inspired by their biological counterparts. They are made of basic units called neurons. Each neuron accepts several inputs, which are then combined (using an activation function) into a single output value. The network is then built by connecting many of these neurons across several layers. Neural networks are nowadays excelling at several artificial perception fields, but with the major disadvantage of being a *black-box* model, which makes it difficult for humans to explain and interpret the reasoning within the network.

ANNs and the term *Deep learning* are not new [20], but it was not until 2006 that the field started to receive renewed interest, when [21] introduced a new fast algorithm to train deep networks. This was seen as a breakthrough. The possibility to train deep models allowed different levels of knowledge for a problem to be learned, with the lower layers learning the low-level details, and the higher layers at the end learning more abstract features. These networks were named Fully Connected neural networks (FCNN) or simply Deep neural networks (DNNs), although they are also referred to as feed-forward networks or multilayer perceptrons (MLP). FCNN is probably the most well known model in deep learning. In FCNNs, neurons are organized in layers, in such a way that neurons in a layer receive inputs from the outputs produced by the previous layer neurons, as figure 2 shows.

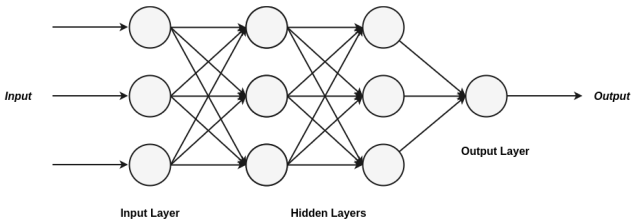


FIGURE 2. Fully Connected Neural Network (FCNN). Information flows from the input layer, through hidden layers, to the output layer. Neurons are distributed in layers. The neurons of each layer are fully connected to the neurons of the following layer.

FCNNs have been used in problems related to audio or music [22] and there is a belief that related MIR problems still have room for improvement with the potential of deep learning [23]. Music analysis challenges can be approached in a multi-modal way and enriched using different sources of information (acoustic features, editorial meta-data or listening stats). This makes the problem richer and more suitable for deep models, which require larger amounts of data. Different types of deep learning models have been applied to audio classification problems, such as Deep Belief Networks (DBNs), Restricted Boltzmann Machines (RBMs), Convolutional neural networks (CNNs) [24] and Recurrent neural networks (RNNs) [25]–[27].

In our experiments, we include a 4-layer FF network, which obtains a mean AP of 0.465 and an AUC of 0.930.

E. CONVOLUTIONAL NEURAL NETWORKS (CNNs)

CNNs are a type of ANN especially suitable for artificial vision, inspired by how neurons are distributed in the biological visual cortex [28]. These models mainly comprise two types of special layers: convolutional and pooling layers. Convolutional layers filter input data to detect certain aspects of the input image, such as borders, corners, lines or other shapes, and eventually learn patterns. Convolutional layers are stacked in a network, in such a way that the lower layers learn the low-level information, whereas the details are learned in the high-level layers.

Convolutional layers provide the ability to detect different patterns in different locations of an image. However, they can lead to overfitting problems as they are sensitive to the location of the detected features. Pooling layers fix this problem by reducing the dimension of the image. This solution adds location invariance capabilities to the network, while reducing the memory and computation requirements. In common CNN architectures, the network comprises pairs of convolutional and pooling layers, followed by a usual feed-forward network. AlexNet [29], VGG [30] and ResNet-50 [31] are commonly used architectures.

Even though CNNs were initially conceived for artificial vision, they have yielded good results in problems in which visual representations of input data are feasible. When working with audio, CNNs are frequently used in combination with mid-level time-frequency visual representations, known as spectrograms [32]. The Audioset VGGish model is an example of this. Although the number of successful applications of this technique is notable [33]–[35], recent studies highlight that ad-hoc models for audio input are avenues also worth pursuing [23], [36].

F. RECURRENT NEURAL NETWORKS (RNNs)

RNNs are focused on representing a sequence of events and extracting knowledge from the sequence. Each neuron in a RNN acts as a feed-forward neuron, except for an additional connection that sends its output back to itself. With each sequence step, the neuron receives two inputs, the input vector at the current step, and the output vector generated by the previous step. With this approach, RNNs are capable of preserving information over time and can work with data of variable length [37]. Examples of use cases are text and speech recognition, translation or music analysis.

A popular architecture of this kind of network is the Long short-term memory (LSTM) network [38], which employs memory cells to enhance the information propagation over time, thus helping the network to discover knowledge through longer sequences. Specifically, LSTM networks include three components: the input gate, the forget gate, and the output gate. These components allow the network to keep information from events further back in time, and therefore make more complex inference decisions based not only on the immediately preceding events.

Because RNNs are well suited to sequential and temporal data, they have been used extensively with audio and music. In the case of MIR topic, RNNs have been used in music analysis [25], music creation [39] or chord recognition [40].

The RNN from our experiments, which is a 3-layer LSTM network, shows a mean AP of 0.437 and AUC of 0.929.

III. STATE OF THE ART

Tzanetakis and Cook were the first to approach the MGC problem in 2002 [5]. The authors used a dataset of 30-feature vectors extracted from audio signals to predict the music genre among 10 different options. Experimenting with the Gaussian mixture model (GMM) and k-nearest neighbors (k-NN) classifiers, the authors achieved accuracy values of 61%. For their study, the authors compiled a dataset, known as the GTZAN dataset, which has eventually become one of the most popular datasets in MGC [41].

The interest and progress in the MGC field is notable, with more than 500 publications reported [42]. Yet, the field still presents open challenges, such as the unclear definition of music genre. Currently, the concept is subject to different perspectives and opinions and vaguely defined [43]. There are several publicly available taxonomies that classify music genres, but they lack agreement on genre definitions and descriptors [43]. Consequently, datasets and data sources normally employed in MGC research do not share a common structure, presenting different classifications and a variable number of genre labels. Moreover, there is no commonly agreed standard dataset for the matter [44], and many of the published research works use private datasets. This circumstance does not enable reproducible results [42].

The most popular dataset, GTZAN, presents defects and inconsistencies [45], which means that the validity of studies based on this dataset has to be assessed with caution. In fact, machine learning approaches that are based just on reproducing ''ground truth'' from different datasets should also be questioned [46]. This is because they can be constructed on an ill-defined concept of music genre. In contrast, more user-centric approaches [47] should be explored.

In recent years, new datasets have been published, generally including a larger number of samples, as well as more diverse sets of features than those included in GTZAN. Audioset, which is used in this study (and covered in the next section), is an example of this. Other popular datasets are the Million Song Dataset (MSD) [48], FMA [49] or MuMu [50]. As new datasets appear, each with its own structure and features, we observe more difficulties in reaching an agreement on genre descriptors [44].

An important research area within MGC is automated feature learning. Feature extraction has traditionally been based on manual and hand-crafted methods, specifically tailored to the undertaken task [18], [41], [51]. Content-based audio features are extracted from raw signals. These features describe the audio in terms of pitch, timbre or rhythm, among other descriptors, and are usually extracted using the Short Time Fourier Transform (STFT) [52]. A commonly

used timbral feature is *Mel-frequency cepstral coefficients* (MFCCs). MFCCs capture the spectrum of sounds and have performed well in different MIR problems [53], [54]. Other common features are *spectral centroid*, *spectral rolloff*, and *Time domain Zero crossings* [5].

Machine learning methods stand as an alternative for learning features automatically, with techniques such as sparse coding algorithms, which have been particularly successful [55], [56]. Recently, deep learning approaches have allowed researchers to handle larger datasets and larger, multi-modal combinations of features [25], [57], [58]. Different network architectures have been used with success for feature learning, such as Deep Belief Networks (DBN) [59], [60], RNNs [25] or CNNs [61].

CNNs normally use audio visual representations (spectrograms) as inputs [32], [62]. After the CNN is trained, it is not uncommon to see how researchers apply a transfer learning approach and extract the embeddings of the final layers from the network to generate pre-trained models [57], [63]. An example of this is Audioset [64], which provides a set of 128 embeddings for each second, learned with a CNN architecture called VGG [30]. Other researchers have experimented with the direct use of raw audio signals as inputs for CNNs, achieving results comparable to spectrogram-based approaches [65] or, given enough data, outperforming spectrogram-based models [36]. RNNs have also been used to learn musical features, given their ability to work with temporal data. [25].

SVMs were the first machine learning models to be used as genre classifiers [66], [67]. More recently, research studies have been experimenting with deep learning models, such as CNNs [68], [69] or RNNs [54]. Human-centric approaches, context-based methods and hybrid classification are being explored as well, proposing solutions that introduce the user as a principal part of the decision process [70], [71].

For a complete review of the state of the art, we refer the reader to our review [6].

IV. TRAINING MODELS WITH AUDIOSET

Even though the details about the experimental training phase are outside the scope of this article, we briefly describe the key parts of the process for clarity. A more detailed explanation about the process and the results is given in [6].

We conducted our experiments with the different models described above. Specifically, we trained a Decision Tree, a Naive Bayes classifier, a linear SVM, a 4-layer feed-forward neural network and a 3-layer LSTM recurrent neural network.

Notice that the distinct models here coded use the best configuration of parameters found empirically at [6]. The hyper-parameter tuning phase was conducted through multiple preliminary training sessions on the Audioset balanced split using 3-fold cross validation.

Given that Audioset samples are each ten second long, we perform the classification in music segments of that length. Specifically, we split each analyzed song in segments of this length and apply the classification model to each of those segments. This technique allows us to classify different parts of a song and deliver classification results in chunks as the song progresses, giving us the ability to provide visual feedback on predominant genres along the song timeline.

A. AUDIOSET

Audioset is an ontology and a dataset of sounds extracted from YouTube [64]. The dataset contains more than 2 million 10-second samples, annotated with entities from the ontology. The ontology tries to cover all types of sound categories, including music, in which it presents a detailed hierarchy of music genres and subgenres. For more details, we refer the interested reader to the Audioset website.[1]

Although Audioset includes audio samples related to many categories, we only use the subset of music genre samples to train the models. This subset is the result of selecting samples tagged with music genre labels present in the ontology. The Audioset ontology includes 52 music genre labels and the dataset is available in three parts: balanced, unbalanced and evaluation sets. After selecting records tagged with genre labels, the balanced set was reduced from 22176 to 2490 samples, and the unbalanced set from 2042985 to 193481 samples, thus leaving us with a music-genre subset that represents approximately 10% of the full dataset.

In the provided dataset, each sample is characterized by a set of 128 features per second, called embeddings, which are the result of feeding the raw audio samples to a model, named by Audioset creators as VGGish,[2] and based on the VGG audio classification model trained on a preliminary version of YouTube-8M dataset [72]. Notice that, in Audioset, the labeling is performed in 10-second extracts, and this is one of the main reasons for having chosen that segment length in our training process, as it makes the models appropriate for that task in the real application presented in the current paper.

Each of the models included in this study is trained using these embeddings, therefore accepting a matrix of 128 x 10 input parameters. Evaluation of each model is performed on the Audioset evaluation split, considering only those samples tagged with music genre tags. In order to be able to deploy the models so that they can classify any music clip, our application includes a conversion layer to perform the conversion from the raw audio signal to a 128-dimensional embedding format.

B. RESULTS

The best performing models are the feed-forward neural network (FF) and the LSTM recurrent neural network. These deep learning models produce results comparable with the Audioset baseline generic classifier study [64], which achieves a mean average precision (AP) of 0.314 and an average AUC score of 0.959. In particular, the FF trained on the unbalanced Audioset yields the best AP / AUC scores

[1] https://research.google.com/audioset/

[2] VGGish model: https://github.com/tensorflow/models/tree/master/research/audioset/vggish

TABLE 1. Experimental results. Each of the models of the experiment is run in the Balanced and Unbalanced dataset, being the initial shown in column T. For each run, we calculated mean average precision (μ_{AP}), mean ROC area under the curve (μ_{AUC}), maximum average precision ($\max_{AP}$), maximum ROC area under the curve ($\max_{AUC}$) and the training time in seconds. Maximum scores also indicate the genre obtaining that score.

Model	T	μ_{AP}	μ_{AUC}	$\max_{AP}$	$\max_{AUC}$	Time (s)
Decision Tree	B	0.060	0.560	0.170 (Music of Bollywood)	0.672 (Ambient music)	18
Decision Tree	U	0.070	0.571	0.194 (Opera)	0.661 (Ambient music)	1799.62
Linear SVM	B	0.126	0.596	0.432 (Music for children)	0.778 (Opera)	10.7
Linear SVM	U	0.107	0.565	0.461 (Music for children)	0.900 (Music for children)	29501
Naive Bayes	B	0.196	0.805	0.436 (Opera)	0.913 (Beatboxing)	2.18
Naive Bayes	U	0.176	0.830	0.358 (Electronic Dance Music)	0.913 (New-age music)	100.7
FF	B	0.417	0.909	0.779 (Music for children)	0.980 (Music for children)	43.77
FF	U	0.465	0.930	0.820 (Music for children)	0.980 (A Capella)	767.486
LSTM	B	0.356	0.890	0.699 (Opera)	0.963 (Trance music)	282.63
LSTM	U	0.437	0.929	0.794 (Music for children)	0.976 (Beatboxing)	5926.59

with values of 0.465 / 0.930. The ability of the NB classifier to accurately predict certain classes (Beatboxing) after only 2.18 seconds of training is also interesting.

The results also present significant performance variations across different genres. Whereas genres such as ''Opera'', ''Flamenco'' or ''Music for children'' are easily detected, other broader genres such as ''Jazz music'' or ''Independent music'' obtain lower scores, and this is reflected as lower scores for all models.

V. DEPLOYING THE MODELS: THE MUSIC GENRE ANALYZER TOOL

A. BACKGROUND: MODELING DECISIONS

Before introducing the design and use of our application, we would like to comment on certain decisions which affect the application and may deserve some discussion.

Firstly, we opted to include YouTube as the audio source, given its undoubted popularity and the immense amount of music it offers. Additionally, it was considered appropriate because the dataset used for training our models, Audioset, included specifically sound files extracted from YouTube. However, this strong and innovative point in our application also involves certain restrictions, as the number of requests we can send to YouTube is limited, due to restrictions in the server. So, two main decisions were made: the length of the excerpts for genre classification is 10 seconds (as this is the way the models were learned) and these excerpts are not overlapped. Many research works cover the configuration of these parameters; some studies even refer to using just a central part in the song to classify a genre, which could differ in length [73]. In previous research works and music datasets, we have found different approaches, that vary in excerpt length, finding cases from 4 to 30 seconds ([73]–[75]). It seems then, that our intermediate approach, 10 seconds, could be an acceptable solution. We wanted to deploy the song in a timeline so that we could visualize the evolution in the genre classification, and this decision for an *average* song gives us around 20-25 chunks. Regarding the overlapping feature, there may exist other systems which use a shifting window of shorter length when applying MGC. Unlike our application, they would be applied to direct audio input, which limits the number of songs that can be analyzed. We could make an alternate application, able to manage song files and perform that kind of analysis, but it will lose the possibility of analyzing any link within the YouTube platform.

Note also that the Decision Tree has not been included in the application. We have decided not to deploy this model, due to the poor performance shown during the experimental training phase.

The application is available at https://jramcast.github.io/mgr-app/. In its front-end, the application resembles what is shown in figure 3, which includes, basically, a brief explanation of the experiment, an input field to specify the song to be analyzed and the results section at the bottom. In the back-end, the application embeds the classification models and exposes an endpoint to use them. For now, the tool only accepts YouTube videos, although it could be extended with other music sources in future stages of our research.

B. INPUT INFORMATION

The only input information required from the user in the front-end side is the unique identifier of a YouTube video, as a URL or a video ID.

In the back-end, the application exposes an endpoint to classify 10-second segments from YouTube videos. The parameters required by this endpoint are the YouTube video ID and the segment start time.

After the user enters the video in the input field and clicks the ''Analyze'' button, the YouTube video starts playing in the background of the main screen and the video identifier is sent to the application back-end to classify the first 10-second segment (from 0:00s to 0:10s). To classify the following segments, a front-end routine is scheduled to call the classification back-end every 10 seconds in coordination with the video playback, specifying the video ID and the segment playing. Theoretically, we could shift the 10-second time window in shorter intervals (so that the second segment could be, for example, from 0:01 to 0:11) and this would deliver a more real-time experience. However, in practice, shorter time shifts generate a higher number of requests to YouTube, which unfortunately results in YouTube rate-limiting our requests.

C. AUDIO PROCESSING AND CLASSIFICATION

When the back-end classification endpoint receives a request, the 10-second audio segment from the specified YouTube video is downloaded in WAV format. Next, the raw WAV

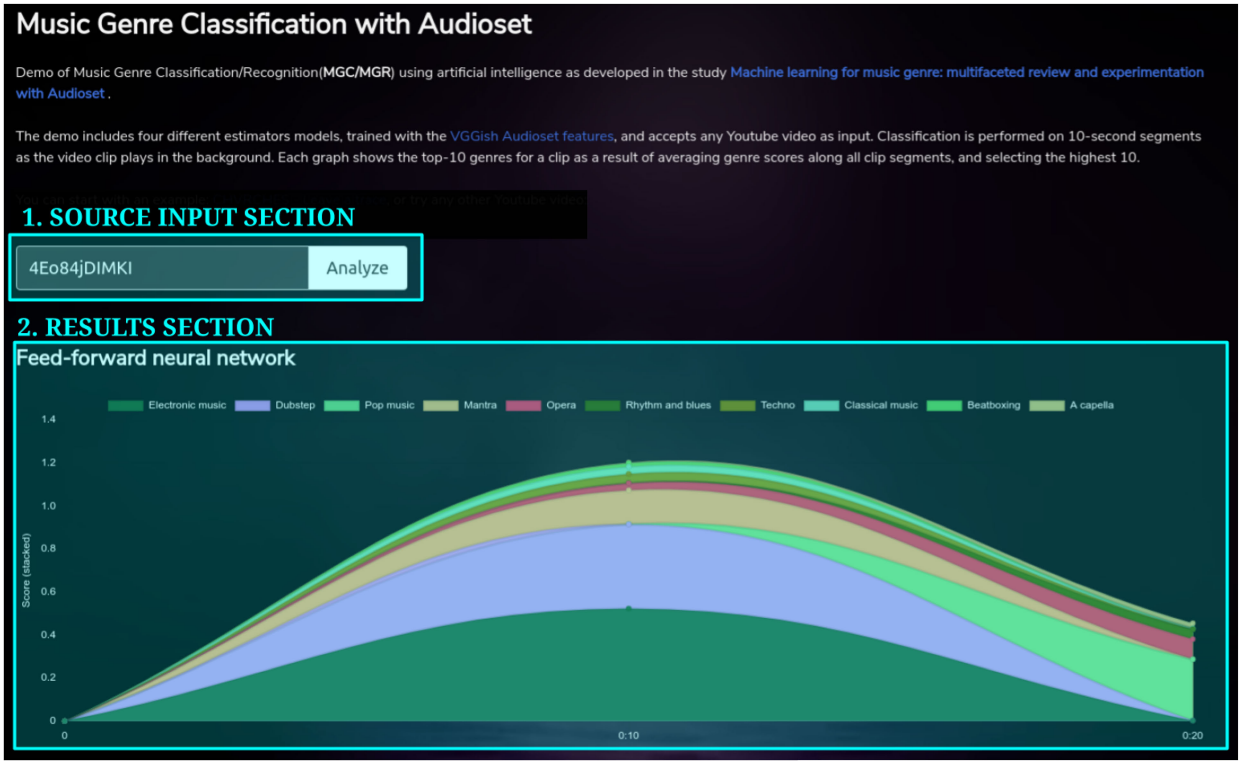


FIGURE 3. Application main screen. The input field in area 1 allows the user to specify a YouTube video, either with a video ID or a URL. The results area (2) shows classification results for each 10-second fragment in the song and for each model. The music video being analyzed is shown in the background.

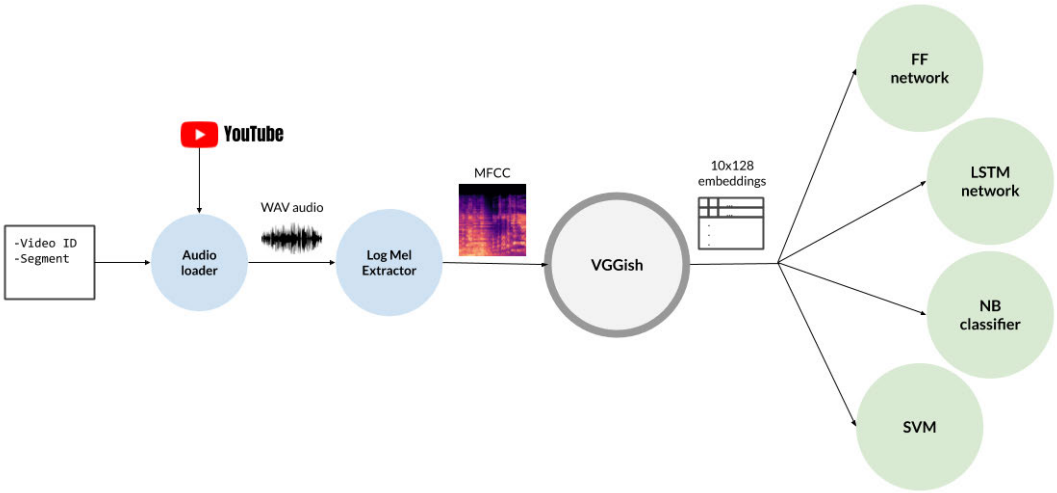


FIGURE 4. Classification flow. Given a video ID and a segment, the audio is downloaded, pre-processed by converting it to MFCCs, then to VGGish embeddings, and finally it is passed to the models for classification.

audio is first converted to Mel Frequency Cepstral Coefficients (MFCCs) [76] and then to 128-dimensional embeddings with the VGGish model. Note this is a necessary step as the models in this research have been trained using these features (see Subsection IV-A). The extracted features are then fed into the models. The data flow of the classification process is as shown in figure 4.

Once outputs are ready for each model, they are combined into a single response object and returned to the front-end, which refreshes the graphs to show the results.

D. OUTPUT INFORMATION

As soon as we start analyzing a clip, the application shows the video in a shadowed background, as well as the result of the four models plotted in stacked area charts. As the song progresses, the result charts are updated showing the classification results associated with the current 10-second excerpt of the song playing, as depicted in figure 5. Music genres shown in each graph correspond to the top-ten genres. The top-10 genres list for the whole song is calculated after every graph refresh. For each genre, we sum all the scores

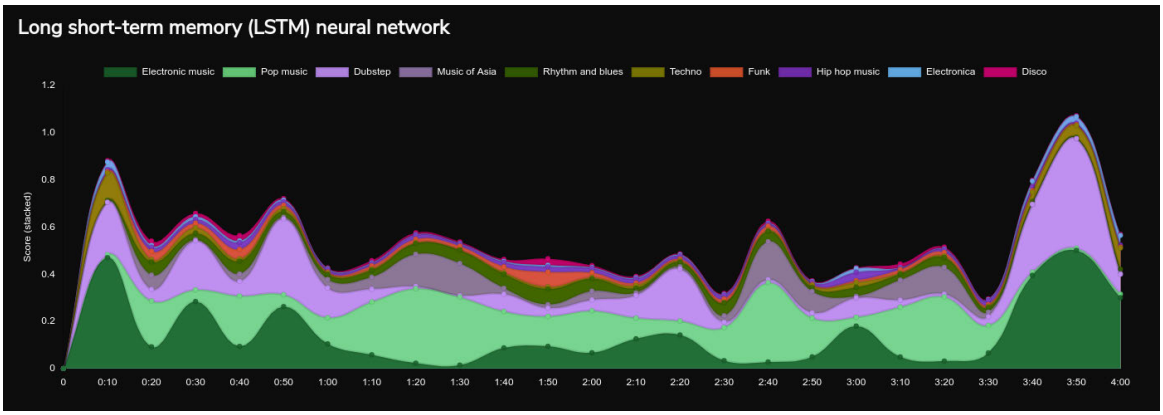


FIGURE 5. Classification output example of the LSTM network for "Leave a trace" by "CHVRCHES".

of the segments analyzed so far and divide the sum by the number of segments analyzed so far. This averaging process also includes the normalization of the scores in the list. The score and its normalized value for each of the top-10 genres are also shown in a list below each graph as a reference.

The front-end receives results from the back-end for each segment in JSON format, including the output for each genre in each of the four studied models. Responses include a result object for each classifier. This object includes a "segment" property containing the video ID, the start and end seconds of the segment, as well as the list of genres and their classification scores, as the example in the listing from figure 6 shows.

```
{
    "FeedForwardNetworkModel": {
        "segment": {
            "fromSecond": 0,
            "mediaUri": "4Eo84jDIMKI",
            "toSecond": 10
        },
        "labels": [
            {
                "name": "Pop music",
                "score": 0.00133908
            },
            ...
        ]
    },
    "LSTMRecurrentNeuralNetwork": { ... },
    "NaiveBayesModel": { ... },
    "SVMModel": { ... },
}
```

FIGURE 6. Classification results format. This format includes information about the segment and the classification scores produced by each classifier.

It should also be considered that the score produced by each classifier does not necessarily have the same meaning or the same scale. Whereas the NB classifier outputs a probability score, the FF classifier and the LSTM classifier return the output of a sigmoid function. This is why we normalize the scores. Normalizing the scores allows us to give a sense of the weight given to each genre and enables comparisons between different models.

VI. USE CASES

We have selected three use cases to analyze and evaluate the application performance. Our intention is to show how the models respond to different music genres across different parts of a song, as well as comparing the resulting genres with the classification on popular online platforms. Because we need some reference ground truth to evaluate the outcomes of the models, we retrieved genre information from three online services: Last.fm,[3] Discogs[4] and Wikipedia,[5] as shown in Table 2.

TABLE 2. Genre categorizations of online services for the three use cases. Last.fm genre top tags have been curated to only include tags referring to genres. Discogs offers a two-level categorization. The first level is called "Genre" and the second level is called "Styles". Discogs styles can be interpreted as subgenres. Wikipedia classifies songs into genres. § → No Wikipedia entry found for this song.

Online service	Queen: Bohemian Rhapsody	Paco de Lucía: Entre dos aguas	Sensible Soccers: AFG
Last.fm (top genre tags)	Classic rock Rock Glam rock Progressive rock Rock opera Pop	Flamenco Instrumental Spanish Chillout Ambient World	Chillout Ambient Post rock Ambient post
Discogs (genres and styles)	*Genres*: Rock *Styles*: Pop Rock	*Genres*: Folk, world & country *Styles*: Flamenco	*Genres*: Electronic Rock *Styles*: Alternative Rock Ambient Experimental Leftfield
Wikipedia (genres)	Progressive rock Hard rock Progressive pop	Flamenco	§

The selected online services provide genre information in different ways. Last.fm offers possibly the largest community database of listening statistics. Discogs is focused on

[3]https://www.last.fm/
[4]https://www.discogs.com/
[5]https://www.wikipedia.org/

editorial and metadata information. Finally, Wikipedia offers articles with genre categorizations for popular or famous artists and songs. Whereas Last.fm relies exclusively on community-contributed tags, Discogs and Wikipedia classify music using their specific genre categorizations. In the case of Discogs, they first classify a song into a ''Genre'' and then add a second-level classification called ''Style'', which they also refer to as a ''sub-Genre''. Last.fm top tags can be retrieved from their API. Each tag includes a name and a weight ranging from 0 to 100, which represents the tag popularity. First, we filtered out the tags not relative to genres. Next, to reduce the amount of irrelevant tags, we filtered out tags with a weight lower than four. We selected this threshold after manually inspecting the Last.fm tags from the three use cases.

A. CASE OF USE 1: BOHEMIAN RHAPSODY BY QUEEN

The first use case refers to ''Bohemian Rhapsody'' by Queen,[6] a popular song that covers multiple genres. Online services show an agreement on Rock and its subgenres as the prevailing genres in this song, mostly in combination with Pop and Opera. The top genre tags in Last.fm for this song are ''Classic rock'', ''Rock'', ''Glam rock'', ''Progressive rock'', ''Rock opera'', and ''Pop''. Discogs considers the song as ''Rock'' and ''Pop Rock''. Finally, the Wikipedia article classifies the song as ''Progressive rock'', ''Hard rock'', and ''Progressive pop''. The genres retrieved from Wikipedia correspond to the article written in English.[7] Interestingly, the Wikipedia genre classification changes across different translations of the same article.

The song opens with a coral vocal fragment that is identified principally as ''Chant'' and ''A capella'' by the FF network, ''Mantra'', ''Vocal Music'' and ''A capella'' by the LSTM network, ''A capella'', ''Vocal'' and ''Gospel'' by the NB classifier, and ''Grunge'' and ''A capella'' by the SVM.

The following part emphasizes vocals and piano and goes in crescendo until the third minute, while gradually incorporating other instruments, such as drums and electric guitars. The FF and LSTM networks detect a mixture of genres in this fragment, such as ''Pop'', ''Blues'', and ''Mantra'' with no clear prevailing genre.

The part in minute four is primarily identified by the FF network as ''Punk rock'', whereas the LSTM network shows higher scores for ''Rock music'' and ''Rock and Roll''. The NB classifier only detects ''Rock and Roll'' in this part. The SVM yields a peculiar result by classifying this part as ''Drum and Bass''.

Towards the end of the song, the FF network shows a combination of ''Classical music'', ''Mantra'' and ''Opera''. The LSTM network believes this section is essentially ''Mantra''. The NB classifier distinguishes ''A capella'', ''Vocal music'' and ''Music of Asia''. The SVM, clearly gives the highest score to ''Classical music'' for this part.

As table 3 shows, after averaging and normalizing scores for each genre across all the song segments, the top-3 genres (and their normalized scores) for the whole song are as follows:

- FF network: ''Chant'' (0.21), ''A capella'' (0.20), and ''Mantra'' (0.18).
- LSTM network: ''Mantra'' (0.20), ''Pop music'' (0.13), and ''Vocal music'' (0.12).
- NB: ''Pop music'' (0.18), ''A capella'' (0.15), and ''Rock and roll'' (0.15).
- SVM: ''Grunge'' (0.41), ''Drum and bass'' (0.12), and ''Music for children'' (0.12).

Both the NB classifier and the SVM classifier were unable to deliver predictions in some fragments of the song.

B. CASE OF USE 2: ENTRE DOS AGUAS BY PACO DE Lucía

''Entre dos aguas'' by Paco de Lucía[8] is a well-known representative song of Flamenco. Online services agree on Flamenco as the main genre for this song, with some common references to World music. Last.fm top genre tags are ''Flamenco'', ''Instrumental'', ''Spanish'', ''Chillout'', ''Ambient'', and ''World''. Discogs considers the song as ''Folk, world & country'' and ''Flamenco''. Wikipedia categorizes this song as ''Flamenco'' as well.

Scores presented in table 3 show the following top 3 song genres (and their normalized scores) for the whole song are as follows:

- FF network: ''Flamenco'' (0.74), ''Music of Latin America'' (0.13), and ''Folk music'' (0.03).
- LSTM network: ''Flamenco'' (0.45), ''Music of Latin America'' (0.18), and ''Jazz'' (0.10).
- NB: ''Flamenco'' (0.39), ''Music of Latin America'' (0.37), and ''Classical music'' (0.09).
- SVM: ''Flamenco'' (0.65), ''Grunge'' (0.16), and ''Music of Latin America'' (0.13).

This use case shows a high consensus across the models. All of them predicted ''Flamenco'' as the main genre with moderately high scores across all the segments. In certain parts of the song, other genres become relevant too, such as ''Music from Latin America'', ''Jazz'' or ''Traditional Music''. The SVM classifier was only able to detect 5 genres across the whole song.

C. CASE OF USE 3: AFG BY SENSIBLE SOCCERS

For the 3rd scenario, we chose ''AFG'' by Sensible Soccers,[9] a less popular song than the previous ones. Last.fm top genre tags are ''Chillout'', ''Ambient'', ''Post rock'', ''Ambient post''. Discogs defines this song as ''Electronic'', ''Rock'', ''Alternative Rock'', ''Ambient'', ''Experimental'' and ''Leftfield''. No articles exist for this song in Wikipedia.

[6]Bohemian Rhapsody by Queen: https://www.youtube.com/watch?v=fJ9rUzIMcZQ

[7]https://en.wikipedia.org/wiki/Bohemian_Rhapsody

[8]Entre dos aguas by Paco de Lucía: https://www.youtube.com/watch?v=DpRb_0IYFV8

[9]AFG by Sensible Soccers: https://www.youtube.com/watch?v=YwVNKMOBiuQ

TABLE 3. Predicted genres and their normalized scores by song and model. Because models use 10-second input fragments, we calculate the genre rankings at the song level by summing the score of each genre across all of the fragments of the song. The application dynamically accumulates the scores of each genre after each fragment, showing the result in the labels list of each graph. The application graphs show genre rankings as legend entries ordered from left to right. The 1st genre is located at the left, whereas the 10th is located at the right. .

Model	Queen: "Bohemian Rhapsody"	Paco de Lucía: "Entre dos aguas"	Sensible Soccers: "AFG"
FF	Chant: 0.21 A capella: 0.20 Mantra: 0.18 Blues: 0.09 Classical music: 0.08 Opera: 0.07 Vocal music: 0.07 Independent music: 0.04 Punk rock: 0.03 Christian music: 0.3	Flamenco: 0.74 Music of Latin America: 0.13 Folk music: 0.03 Traditional music: 0.03 Country: 0.02 Opera: 0.01 Swing music: 0.01 Blues: 0.01 Classical music: 0.01 Middle Eastern music: 0.01	Mantra: 0.23 Classical music: 0.21 Electronic music: 0.12 Traditional music: 0.09 Punk rock: 0.07 Opera: 0.06 Blues: 0.06 Ambient music: 0.06 Independent music: 0.05 Chant: 0.05
LSTM	Mantra: 0.20 Pop music: 0.13 Vocal music: 0.12 Rock and roll: 0.11 Rock music: 0.09 Electronic music: 0.08 Gospel music: 0.07 A capella: 0.07 Christian music: 0.07 Rhythm and blues: 0.07	Flamenco: 0.45 Music of Latin America: 0.18 Jazz: 0.10 Blues: 0.06 Electronic music: 0.04 Swing music: 0.04 Country: 0.03 Funk: 0.03 Music of Africa: 0.03 Folk music: 0.03	Electronic music: 0.40 Techno: 0.12 Ambient music: 0.08 Dubstep: 0.07 Funk: 0.06 Electronica: 0.06 Rock and roll: 0.06 Rock music: 0.06 Classical music: 0.05 Pop music: 0.05
NB	Pop music: 0.18 A capella: 0.15 Rock and roll: 0.15 Vocal music: 0.14 Rock music: 0.12 Music of Asia: 0.10 Blues: 0.07 Gospel music: 0.06 Jazz: 0.02 Christian music: 0.01	Flamenco: 0.39 Music of Latin America: 0.37 Classical music: 0.09 Country: 0.06 Traditional music: 0.05 Bluegrass: 0.04 Swing music: 0.01 Folk music: $2.9e$-4 Jazz: $1e$-4 Blues: $5.8e$-4	Rock music: 0.18 Rock and roll: 0.17 Electronic music: 0.16 Pop music: 0.10 Independent music: 0.08 Funk: 0.07 Blues: 0.07 Jazz: 0.06 Hip hop music: 0.05 Grunge: 0.03
SVM	Grunge: 0.41 Drum and bass: 0.12 Music for children: 0.12 A capella: 0.12 Blues: 0.11 Classical music: 0.06 Electronic music: 0.03 Mantra: 0.02 Vocal music: 0.01	Flamenco: 0.65 Grunge: 0.16 Music of Latin America: 0.13 Blues: 0.04 Disco: 0.02	Electronic music: 0.44 Ambient music: 0.13 Grunge: 0.10 Drum and bass: 0.09 Blues: 0.07 Mantra: 0.08 Independent music: 0.04 Classical music: 0.02 Hip hop music: 0.01 Middle Eastern music: $2.3e$-3

In the first part of the song, until minute 5, the FF network outputs ''Classical music'', ''Mantra'' and ''Electronic music''. The LSTM, potentially identifies the song as ''Electronic music'', ''Techno'', and ''Dubstep'' as predominant genres. The Naive Bayes is unable to predict any genres, except for ''Electronic music'' in some sparse segments. The SVM predicts ''Electronic music'', ''Drum and bass'' and ''Ambient music'', also sparingly.

In the second part of the song, with greater predominance of the electric guitar, the FF network detects a significant influence of ''Punk rock'' and ''Independent ''music''. The LSTM network detects a moderate presence ''Rock music''. The NB classifier becomes more responsive in this part, detecting ''Rock and roll'', ''Rock music'' and ''Independent music''. The SVM detects ''Grunge'' and ''Drum and bass''.

The third and closing part comes back to the predominance of ethereal synthesizer sounds. The FF network assigns high scores to ''Mantra'' again and outputs ''Ambient music'' with a relevant score. The LSTM rises the score for ''Electronic music'' and also predicts ''Ambient music'' with more confidence. The NB classifier is unable to generate relevant predictions. Finally, the SVM outputs a combination of ''Electronic music'', ''Ambient'', and ''Mantra''.

At the song level, the average and normalized scores presented in table 3 show the following top-3 song genres (and their normalized scores) for the whole song are as follows:

- FF network: ''Mantra'' (0.23), ''Classical music:'' (0.21), and ''Electronic music'' (0.12).
- LSTM network: ''Electronic music:'' (0.40), ''Techno'' (0.12), and ''Ambient music'' (0.08).

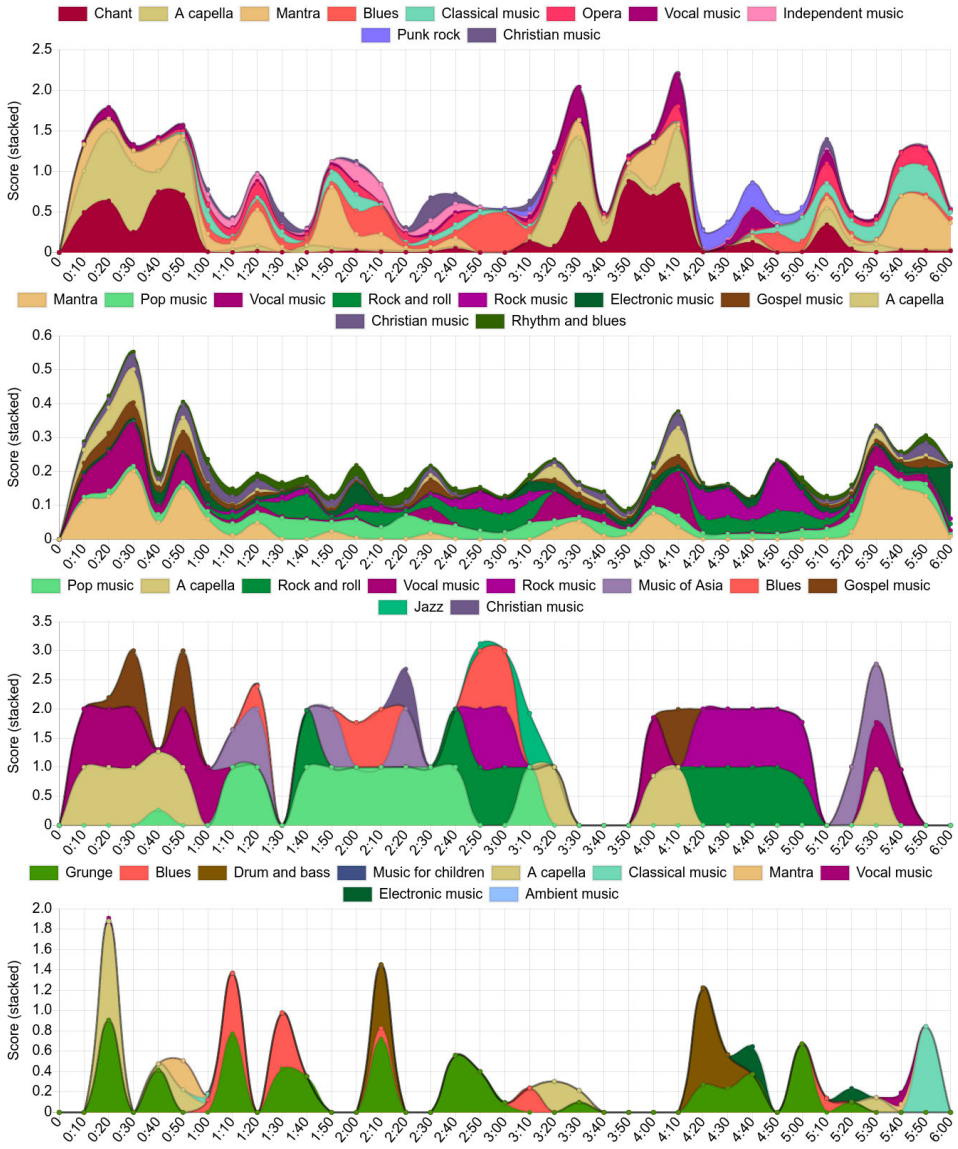


FIGURE 7. Top 10 genres for Queen - Bohemian Rhapsody. From top to bottom: FF network, LSTM network, NB, and SVM.

- NB: "Rock music" (0.18), "Rock and roll" (0.17), and "Electronic music" (0.16).
- SVM: "Electronic music" (0.44), "Ambient music" (0.13), and "Grunge" (0.10).

We also notice that, although the video is silent from 9:20, all classifiers, except for the SVM classifier, detect genres in this part.

VII. EVALUATION AND DISCUSSION OF THE RESULTS

A. SONG-LEVEL EVALUATION AND METRICS

The evaluation of results can be carried out at two levels: the song level and the segment level. To evaluate the performance at the song level, we have decided to compare the genre annotations produced by our application against the genres provided in popular online music data services.

What we propose for this article is a simple compound metric that measures the precision and the sensitivity. The precision measures the degree to which predicted genres are present in online services. The sensitivity measures how well genres present in online services are detected by the application.

The main problem we face when comparing predicted genres with the information in online services is how to identify a positive (and a negative) match. Note that we are trying to match the genres of different categorical representations. Representations from Last.fm, Discogs and Wikipedia are different from each other and also differ from the Audioset ontology [77]. This is a well-known problem in MGC [46]. There is not a generally agreed formal definition for music genre. The available categorizations are defined using

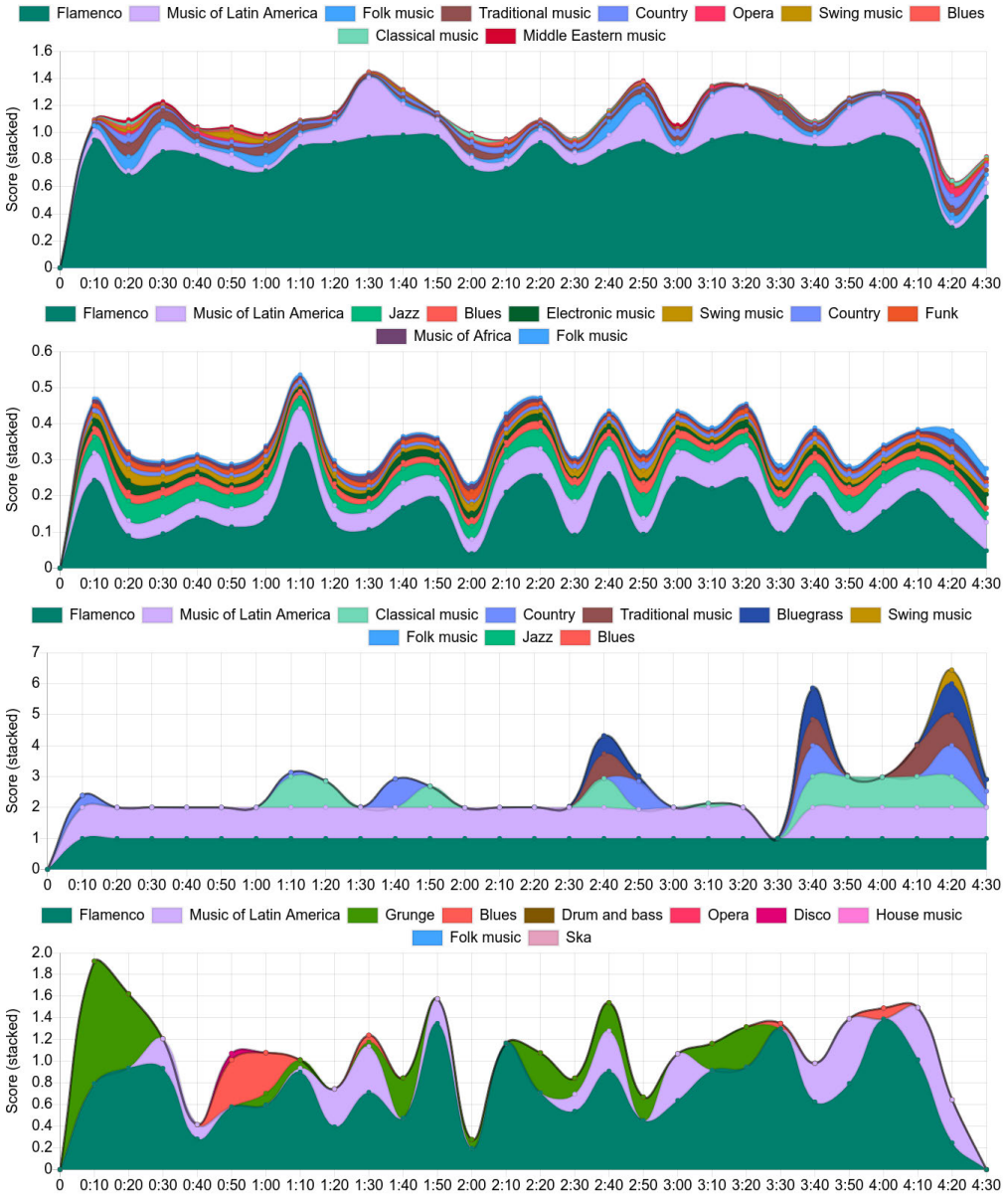


FIGURE 8. Top 10 genres for Paco de Lucía - Entre dos aguas. From top to bottom: FF network, LSTM network, NB, and SVM.

arbitrary perspectives (e.g. music features, editorial categorization, emotional or cultural effects), genre definitions change or overlap, and levels in hierarchical taxonomies are confusing [78].

Although all these different genre categorizations show little consensus, they are not completely different disjoint sets. Mapping music genres between different sources is so complex that CNNs themselves have been applied to solve this task [79]. Our proposal is to identify the intersection between these categorizations, trying to maximize it to enable the comparison between them.

To simplify the calculation of our metric, we have combined the genres of online services so that we only have to compare between two populations: Audioset genres and online services (in our case, genres from Last.fm, Discogs and Wikipedia combined). Therefore, when computing the metrics, we match the Audioset genres against the online services genres. Our matching algorithm performs a case-insensitive match, subject to the following rules:

- If an exact match is found between the Audioset genres and the online service genres, we count it as one positive match (1). As an exception, we consider that genres that only differ on the "music" suffix are also a match (e.g. "Rock music" and "Rock" are an exact match because both refer to the Rock genre).

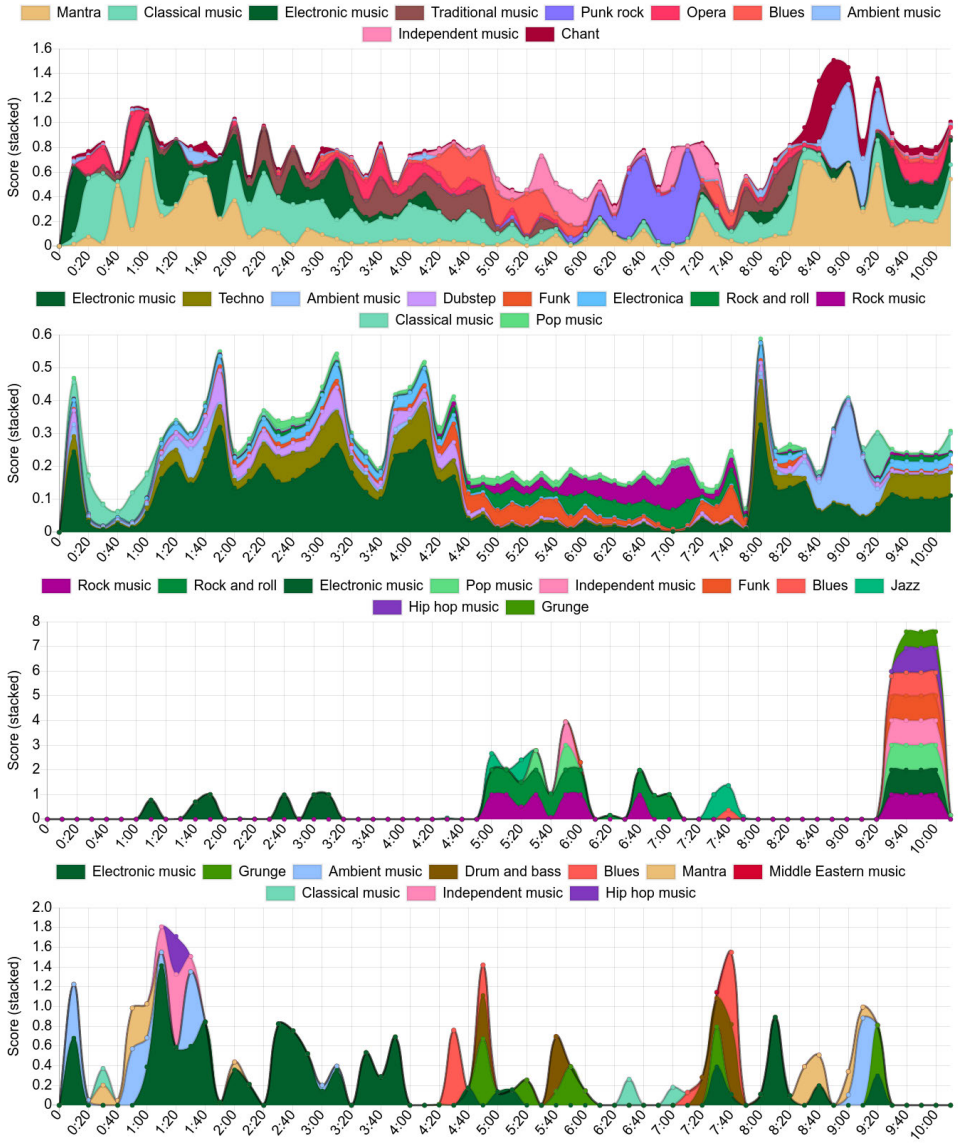


FIGURE 9. Top 10 genres for Sensible Soccers - AFG. From top to bottom: FF network, LSTM network, NB, and SVM.

- Partial matches count as half a positive value (0.5) (e.g. ''Opera'' and ''Opera rock'' both refer in some degree to the Opera genre). Partial matches allow us to detect parent, child or sibling genres).

We call our metric **p/s**, an abbreviation for precision/sensitivity. The precision shows the percentage of predicted genres that are present in online services, whereas the sensitivity measures how well genres from online services are detected by the models.

Precision is computed as $\mathbf{p} = \frac{\sum_{i=1}^{n} r_i}{n}$, where $r_i \in \{0, 0.5, 1.0\}$ and $n = 10$. Notice that r_i represents the match relevance, being 0 if the genres cannot be matched, 0.5 if they match partially (for instance 'Pop' and 'Punk Pop'), and 1.0 if they are equal. We have also included a weighted version. The weighted precision metric gives a sense of how well the predicted genres, considering their prediction scores, match the genres from online services. It is calculated as $\mathbf{p}_w = \sum_{i=1}^{n} r_i \times score_i$, where the new term $score_i$ indicates the normalized score for genre i given by the corresponding model.

Sensitivity is computed as $\mathbf{s} = \frac{\sum_{i=1}^{n} r_i}{n}$, where $r_i \in \{0, 0.5, 1.0\}$ and n is the size of the set made of Last.fm tags, Discogs genres and Wikipedia genres for song i. The computation of r_i follows the same rules used for the precision. The sensitivity score does not have a weighted version because not all online services provide weights (only Last.fm).

The results of evaluating the use cases with the p/s metric are shown in Table 4.

For ''Queen - Bohemian Rhapsody'', the best performing model is the NB classifier, achieving a regular precision value of 0.30 and a weighted precision value of 0.45.

TABLE 4. Precision/Sensitivity matrix. Each cell shows two values: precision, which measures how predicted genres match genres in online services, and sensitivity, which measures how well genres from online services are detected by the application. The weighted p/s metric only affects the precision values, because weights have not been considered for genres from external services.

Model	Queen: "Bohemian Rhapsody"	Paco de Lucía "Entre dos aguas"	Sensible Soccers "AFG"
		p/s	
FF	0.90 / 0.41	0.20 / 0.17	0.25 / 0.44
LSTM	0.25 / 0.68	**0.30 / 0.28**	**0.40 / 0.50**
NB	**0.30 / 0.68**	**0.30 / 0.28**	0.25 / 0.33
SVM	0.00 / 0.00	0.15 / 0.17	0.20 / 0.28
		Weighted p/s	
FF	0.05 / 0.41	**0.77 / 0.17**	0.21 / 0.44
LSTM	0.27 / 0.68	0.59 / 0.28	**0.59 / 0.50**
NB	**0.45 / 0.68**	0.42 / 0.28	0.43 / 0.33
SVM	0.00 / 0.00	0.66 / 0.17	0.58 / 0.28

The NB classifier, along with the LSTM network, achieves the highest sensitivity. It is also worth mentioning that the SVM does not detect any correct genres in this use case.

In ''Paco de Lucia - Entre dos aguas'', the NB classifier and the LSTM network achieve a regular precision value of 0.30. These two classifiers yield the highest sensitivity with a value of 0.28. When using the weighted precision, the FF network achieves a value of 0.77.

Finally, for ''Sensible Soccers - AFG'', the LSTM network achieves the best results, with a regular precision value of 0.40 and a weighted precision value of 0.59. This model also achieves the best sensitivity, with a value of 0.5.

B. SEGMENT-LEVEL DISCUSSION

Despite the best of our efforts, we have been unable to find a generally available database with genre annotations at the segment level. Nevertheless, we believe we can still open a discussion and draw some conclusions from how each model behaves.

In the first use case, the model that gives the closest results to existing online categorizations is Naive Bayes, which resembles labels from Last.fm, Wikipedia and Discogs. We noted that the FF network has a tendency to predict ''Mantra'' in many occasions. In general, deep learning methods show outputs comprising a combination of genres. The NB classifier and the SVM, in contrast, are more restrictive in their results, showing no more than four or five positive labels at once. Also, when predicting positive labels, these methods present much more confidence or probability in their scores than deep learning methods.

In the second use case, where the song presents a lower degree of genre fusion, being in fact a classic Flamenco piece, models show more stable and confident results. In this case, all of them predict ''Flamenco'' as the top rated genre, with a much higher score than other genres in the top ten, with the exception of NB, which also includes ''Music of Latin America'' as a highly probable genre for the whole song. This is a clear case of agreement between categorizations and the results from our models.

The third use case uses a song that again mixes different electronic genres. Being, in the broad sense, a piece of electronic music, it also includes, especially towards the second half of the song, features that are usual indicators of other genres such as heavy use of electric guitars and distortion. In this case, we believe that the LSTM gives a more stable prediction, which seems closer to reality. The FF network shows its tendency towards the ''Mantra'' genre or even classical music. We could argue that the song has some touches from these genres, but we definitely believe this is not a ''Mantra'' or ''Classical music'' song. NB faces a challenge here, as it is not able to detect any genres in many parts of the song. The SVM, however, shows better results in this case. The last 40 seconds of the song are mostly silent, and it is interesting how all models, except for the SVM, consider that the silence corresponds to several genres. It is also interesting how the SVM, which produces bad results in the first 2 scenarios, gives better predictions for this case. In general, the models were able to agree with online categorizations with regard to the ''Electronic music'' genre, whereas ''Ambient'' and ''Rock music'' were only partially detected.

VIII. CONCLUSION

The article presents a web application to discover music genres present in a song, along its timeline, based on a previous experimentation with different machine learning models [6]. By identifying genres in each 10-second fragment, we can get an idea of how each model perceives each part of a song. Moreover, by presenting those data in a stacked area timeline graph, the application is also able to quickly show the behavior of the models, which at the same time, is an interesting way to detect undesired or rare predictions.

We believe that this application could be a supporting tool for the traditional evaluation metrics in MGC, especially when manual introspection of questionable results is required beyond classic performance metrics, such as average precision or AUC.

It is, in any case, a challenge to establish a formal way to validate genre predictions, particularly when trying to compare them with categorizations from other sources, such as online music platforms, because there is no standard or formal way of defining genres. Last.fm, to name an example, has a completely different set of tags, which, in many cases, do not correspond or exist in the Audioset ontology.

The application is also a first step towards an eventual user-centered MGC tool, in which the users can submit feedback about the correctness of the predictions. To our knowledge, there is no visual tool that provides this level of verification on genre classification results for different fragments of the song.

The design of the precision/sensitivity metric, and its use for comparing the models' results, is an additional contribution of this paper. The incorporation of available tags from public and online services enabled the proposed evaluation method. We believe that the extension and refinement of these metrics and matching algorithms is a promising future line of

work and deserves attention. As mentioned throughout the paper, a consensus for a standardized taxonomy for music genre categorization is an open challenge for MGC. We plan to open a research line approaching this issue, and we feel we should incorporate semantic elements and ontology-based information to properly tackle the genre-mapping problem across different taxonomies.

REFERENCES

[1] J. S. Downie, "Music information retrieval," *Annu. Rev. Inf. Sci. Technol.*, vol. 37, no. 1, pp. 295–340, 2003.

[2] C.-Z. A. Huang, C. Hawthorne, A. Roberts, M. Dinculescu, J. Wexler, L. Hong, and J. Howcroft, "The bach doodle: Approachable music composition with machine learning at scale," 2019, *arXiv:1907.06637*. [Online]. Available: http://arxiv.org/abs/1907.06637

[3] J.-P. Briot, G. Hadjeres, and F.-D. Pachet, "Deep learning techniques for music generation—A survey," 2017, *arXiv:1709.01620*. [Online]. Available: http://arxiv.org/abs/1709.01620

[4] H. Li, "Piano automatic computer composition by deep learning and blockchain technology," *IEEE Access*, vol. 8, pp. 188951–188958, 2020.

[5] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Trans. Speech Audio Process.*, vol. 10, no. 5, pp. 293–302, Jul. 2002.

[6] J. Ramírez and M. J. Flores, "Machine learning for music genre: Multifaceted review and experimentation with audioset," *J. Intell. Inf. Syst.*, vol. 59, pp. 469–499, Nov. 2019.

[7] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1026–1034.

[8] R. Basili, A. Serafini, and A. Stellato, "Classification of musical genre: A machine learning approach," in *Proc. 5th ISMIR Conf.*, Barcelona, Spain, 2004.

[9] J.-J. Aucouturier, F. Pachet, P. Roy, and A. Beurivé, "Signal + context= better classification," in *Proc. 8th ISMIR Conf.*, Vienna, Austria, 2007, pp. 425–430.

[10] T. D. Nielsen and F. V. Jensen, *Bayesian Networks and Decision Graphs*. New York, NY, USA: Springer, 2009.

[11] M. J. Flores, J. A. Gámez, and A. M. Martínez, "Supervised classification with Bayesian networks: A review on models and applications," in *Intelligent Data Analysis for Real-Life Applications: Theory and Practice*. Hershey, PA, USA: IGI Global, 2012, pp. 72–102.

[12] D. Temperley, "A unified probabilistic model for polyphonic music analysis," *J. New Music Res.*, vol. 38, no. 1, pp. 3–18, Mar. 2009.

[13] J. Pickens, "A comparison of language modeling and probabilistic text information retrieval approaches to monophonic music retrieval," in *Proc. 1st ISMIR Conf.*, Plymouth, MA, USA, 2000, pp. 1–11.

[14] H.-S. Park, J.-O. Yoo, and S.-B. Cho, "A context-aware music recommendation system using fuzzy Bayesian networks with utility theory," in *Proc. Int. Conf. Fuzzy Syst. Knowl. Discovery*. Berlin, Germany: Springer, 2006, pp. 970–979.

[15] K. Yoshii, M. Goto, K. Komatani, T. Ogata, and H. G. Okuno, "An efficient hybrid music recommender system using an incrementally trainable probabilistic generative model," *IEEE Trans. Audio, Speech, Language Process.*, vol. 16, no. 2, pp. 435–447, Feb. 2008.

[16] S. A. Abdallah, "Towards music perception by redundancy reduction and unsupervised learning in probabilistic models," Ph.D. dissertation, Dept. Electron. Eng., Queen Mary Univ. London, London, U.K., 2002.

[17] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining Knowl. Discovery*, vol. 2, no. 2, pp. 121–167, 1998.

[18] T. Li, M. Ogihara, and Q. Li, "A comparative study on content-based music genre classification," in *Proc. 26th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr. (SIGIR)*, 2003, pp. 282–289.

[19] C. N. Silla, A. L. Koerich, and C. A. A. Kaestner, "Improving automatic music genre classification with hybrid content-based feature vectors," in *Proc. ACM Symp. Appl. Comput. (SAC)*, 2010, pp. 1702–1707.

[20] R. Dechter, "Learning while searching in constraint-satisfaction-problems," in *Proc. 5th Nat. Conf. Artif. Intell.* Philadelphia, PA, USA: Morgan Kaufmann, 1986, pp. 178–185.

[21] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, Jul. 2006.

[22] L. Deng and D. Yu, "Deep learning: Methods and applications," *Found. Trends Signal Process.*, vol. 7, nos. 3–4, pp. 197–387, Jun. 2014.

[23] E. J. Humphrey, J. P. Bello, and Y. LeCun, "Feature learning and deep architectures: New directions for music informatics," *J. Intell. Inf. Syst.*, vol. 41, no. 3, pp. 461–481, Dec. 2013.

[24] W. W. Y. Ng, W. Zeng, and T. Wang, "Multi-level local feature coding fusion for music genre recognition," *IEEE Access*, vol. 8, pp. 152713–152727, 2020.

[25] S. Böck, F. Krebs, and G. Widmer, "Joint beat and downbeat tracking with recurrent neural networks," in *Proc. 17th ISMIR Conf.*, New York City, NY, USA, 2016, pp. 255–261.

[26] Y. M. G. Costa, L. S. Oliveira, and C. N. Silla, "An evaluation of convolutional neural networks for music classification using spectrograms," *Appl. Soft Comput.*, vol. 52, pp. 28–38, Mar. 2017.

[27] R. Yang, L. Feng, H. Wang, J. Yao, and S. Luo, "Parallel recurrent convolutional neural networks-based music genre classification method for mobile devices," *IEEE Access*, vol. 8, pp. 19629–19637, 2020.

[28] D. H. Hubel and T. N. Wiesel, "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex," *J. Physiol.*, vol. 160, no. 1, pp. 106–154, Jan. 1962.

[29] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[30] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. 3rd Int. Conf. Learn. Represent.*, San Diego, CA, USA, 2015, pp. 1–14.

[31] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[32] K. W. Cheuk, H. Anderson, K. Agres, and D. Herremans, "NnAudio: An on-the-Fly GPU audio to spectrogram conversion toolbox using 1D convolutional neural networks," *IEEE Access*, vol. 8, pp. 161981–162003, 2020.

[33] G. Korvel, P. Treigys, G. Tamulevicus, J. Bernataviciene, and B. Kostek, "Analysis of 2D feature spaces for deep learning-based speech recognition," *J. Audio Eng. Soc.*, vol. 66, no. 12, pp. 1072–1081, Dec. 2018.

[34] S. Gururani, C. Summers, and A. Lerch, "Instrument activity detection in polyphonic music using deep neural networks," in *Proc. 19th ISMIR Conf.*, Paris, France, 2018, pp. 569–576.

[35] J. S. Gómez, J. Abeßer, and E. Cano, "Jazz solo instrument classification with convolutional neural networks, source separation, and transfer learning," in *Proc. 19th ISMIR Conf.*, Paris, France, 2018, pp. 577–584.

[36] J. Pons, O. Nieto, M. Prockup, E. M. Schmidt, A. F. Ehmann, and X. Serra, "End-to-end learning for music audio tagging at scale," in *Proc. 19th ISMIR Conf.*, Paris, France, 2018, pp. 637–644.

[37] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," in *Proc. Deep Learn. Represent. Learn. Workshop*, 2014, pp. 1–10.

[38] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[39] C.-C.-J. Chen and R. Miikkulainen, "Creating melodies with evolving recurrent neural networks," in *Proc. Int. Joint Conf. Neural Netw.*, vol. 3, Jul. 2001, pp. 2241–2246.

[40] N. Boulanger-Lewandowski, Y. Bengio, and P. Vincent, "Audio chord recognition with recurrent neural networks," in *Proc. 14th ISMIR Conf.*, 2013, pp. 335–340.

[41] M. Längkvist, L. Karlsson, and A. Loutfi, "A review of unsupervised feature learning and deep learning for time-series modeling," *Pattern Recognit. Lett.*, vol. 42, pp. 11–24, Jun. 2014.

[42] B. L. Sturm, "A survey of evaluation in music genre recognition," in *Proc. Int. Workshop Adapt. Multimedia Retr.* Copenhagen, Denmark: Springer, 2012, pp. 29–66.

[43] J.-J. Aucouturier and F. Pachet, "Representing musical genre: A state of the art," *J. New Music Res.*, vol. 32, no. 1, pp. 83–93, Mar. 2003.

[44] H. Pálmason, B. T. Jónsson, L. Amsaleg, M. Schedl, and P. Knees, "On competitiveness of nearest-neighbor-based music classification: A methodological critique," in *Proc. Int. Conf. Similarity Search Appl.* Munich, Germany: Springer, 2017, pp. 275–283.

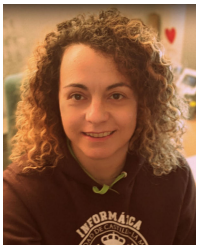
[45] B. L. Sturm, "An analysis of the GTZAN music genre dataset," in *Proc. 2nd Int. ACM Workshop Music Inf. Retr. User-Centered Multimodal Strategies (MIRUM)*, 2012, pp. 7–12.

[46] B. L. Sturm, "The state of the art ten years after a state of the art: Future research in music information retrieval," *J. New Music Res.*, vol. 43, no. 2, pp. 147–172, Apr. 2014.

[47] M. Schedl, A. Flexer, and J. Urbano, "The neglected user in music information retrieval research," *J. Intell. Inf. Syst.*, vol. 41, no. 3, pp. 523–539, Dec. 2013.

[48] B. McFee, T. Bertin-Mahieux, D. P. W. Ellis, and G. R. G. Lanckriet, "The million song dataset challenge," in *Proc. 21st Int. Conf. Companion World Wide Web (WWW Companion)*, 2012, pp. 909–916.
[49] M. Defferrard, K. Benzi, P. Vandergheynst, and X. Bresson, "FMA: A dataset for music analysis," in *Proc. 18th ISMIR Conf.*, Suzhou, China, 2017, pp. 316–323.
[50] S. Oramas, O. Nieto, F. Barbieri, and X. Serra, "Multi-label music genre classification from audio, text and images using deep features," in *Proc. 18th ISMIR Conf.*, Suzhou, China, 2017, pp. 23–30.
[51] P. Herrera-Boyer, G. Peeters, and S. Dubnov, "Automatic classification of musical instrument sounds," *J. New Music Res.*, vol. 32, no. 1, pp. 3–21, Mar. 2003.
[52] L. R. Rabiner and B.-H. Juang, *Fundamentals of Speech Recognition*, vol. 14. Upper Saddle River, NJ, USA: Prentice-Hall, 1993.
[53] M. I. Mandel and D. Ellis, "Song-level features and support vector machines for music classification," in *Proc. 6th ISMIR Conf.*, London, U.K., 2005, pp. 594–599.
[54] K. H. Wong, C. P. Tang, K. L. Chui, Y. K. Yu, and Z. Zeng, "Music genre classification using a hierarchical long short term memory (LSTM) model," in *Proc. 3rd Int. Workshop Pattern Recognit.*, Jul. 2018, pp. 334–340.
[55] E. C. Smith and M. S. Lewicki, "Efficient auditory coding," *Nature*, vol. 439, no. 7079, pp. 978–982, Feb. 2006.
[56] H. Lee, P. Pham, Y. Largman, and A. Y. Ng, "Unsupervised feature learning for audio classification using convolutional deep belief networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2009, pp. 1096–1104.
[57] K. Choi, G. Fazekas, M. B. Sandler, and K. Cho, "Transfer learning for music classification and regression tasks," in *Proc. 18th ISMIR Conf.*, Suzhou, China, 2017, pp. 141–149.
[58] R. Delbouys, R. Hennequin, F. Piccoli, J. Royo-Letelier, and M. Moussallam, "Music mood detection based on audio and lyrics with deep neural net," in *Proc. 19th ISMIR Conf.*, Paris, France, 2018, pp. 370–375.
[59] P. Hamel and D. Eck, "Learning features from music audio with deep belief networks," in *Proc. 11th ISMIR Conf.*, Utrecht, The Netherlands, 2010, pp. 339–344.
[60] E. M. Schmidt and Y. Kim, "Learning rhythm and melody features with deep belief networks," in *Proc. 14th ISMIR Conf.*, Curitiba, Brazil, 2013, pp. 21–26.
[61] Y. Wu and T. Lee, "Reducing model complexity for DNN based large-scale audio classification," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2018, pp. 331–335.
[62] A. Van den Oord, S. Dieleman, and B. Schrauwen, "Deep content-based music recommendation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 2643–2651.
[63] R. Hennequin, J. Royo-Letelier, and M. Moussallam, "Audio based disambiguation of music genre tags," in *Proc. 19th ISMIR Conf.*, Paris, France, 2018, pp. 645–652.
[64] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2017, pp. 776–780.
[65] S. Dieleman and B. Schrauwen, "End-to-end learning for music audio," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2014, pp. 6964–6968.
[66] M. Henaff, K. Jarrett, K. Kavukcuoglu, and Y. LeCun, "Unsupervised learning of sparse features for scalable audio classification," in *Proc. 12th ISMIR Conf.*, Miami, FL, USA, 2011, pp. 681–686.
[67] S. R. Ness, A. Theocharis, G. Tzanetakis, and L. G. Martins, "Improving automatic music tag annotation using stacked generalization of probabilistic SVM outputs," in *Proc. 17th ACM Int. Conf. Multimedia (MM)*, 2009, pp. 705–708.
[68] C. Senac, T. Pellegrini, F. Mouret, and J. Pinquier, "Music feature maps with convolutional neural networks for music genre classification," in *Proc. 15th Int. Workshop Content-Based Multimedia Indexing*, Jun. 2017, pp. 19–23.
[69] U. Marchand and G. Peeters, "The modulation scale spectrum and its application to rhythm-content description," in *Proc. 17th Int. Conf. Digit. Audio Effects*, 2014, pp. 167–172.
[70] P. Knees and M. Schedl, "A survey of music similarity and recommendation from music context data," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 10, no. 1, pp. 1–21, Dec. 2013.
[71] N. Koenigstein, G. Dror, and Y. Koren, "Yahoo! Music recommendations: Modeling music ratings with temporal dynamics and item taxonomy," in *Proc. 5th ACM Conf. Recommender Syst. (RecSys)*, 2011, pp. 165–172.
[72] S. Hershey, S. Chaudhuri, D. P. W. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold, M. Slaney, R. J. Weiss, and K. Wilson, "CNN architectures for large-scale audio classification," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2017, pp. 131–135.
[73] E. Guaus, "Audio content processing for automatic music genre classification: Descriptors, databases, and classifiers," Ph.D. dissertation, Dept. Inf. Commun. Technol., Universitat Pompeu Fabra, Barcelona, Spain, 2009.
[74] A. J. H. Goulart, R. C. Guido, and C. D. Maciel, "Exploring different approaches for music genre classification," *Egyptian Informat. J.*, vol. 13, no. 2, pp. 59–63, Jul. 2012.
[75] N. M. Norowi, S. Doraisamy, and R. Wirza, "Factors affecting automatic genre classification: An investigation incorporating non-western musical forms," in *Proc. Int. Conf. Music Inf. Retr.*, 2005, pp. 13–20.
[76] B. Logan, "Mel frequency cepstral coefficients for music modeling," in *Proc. 1st ISMIR Conf.*, Plymouth, MA, USA, 2000, pp. 1–11.
[77] H. Schreiber, "Genre ontology learning: Comparing curated with crowd-sourced ontologies," in *Proc. ISMIR*, 2016, pp. 400–406.
[78] F. Fabbri, "Browsing music spaces: Categories and the musical mind," in *Proc. Int. Assoc. Study Popular Music*, 1999, pp. 49–62.
[79] S. Panwar, A. Das, M. Roopaei, and P. Rad, "A deep learning approach for mapping music genres," in *Proc. 12th Syst. Syst. Eng. Conf. (SoSE)*, Jun. 2017, pp. 1–5.

JAIME RAMIREZ CASTILLO received the M.Sc. degree in computer science from the University of Castilla-La Mancha, in 2007, where he is currently pursuing the Ph.D. degree. Since then, he has worked as a Software Engineer at different companies. He is currently working as a Curriculum Content Architect at Red Hat. He is a Member of the Intelligent Systems and Data Mining research group in the Albacete Research Institute of Informatics (I3A). His interests include deep learning, probabilistic models, data mining, and music information retrieval.



M. JULIA FLORES received the M.Sc. and Ph.D. degrees in computer science from the University of Castilla-La Mancha, Spain, in 2000 and 2005, respectively. She is a founder member of the research group Intelligent Systems and Data Mining lab in the Albacete Research Institute of Informatics (I3A). Her main research lines are probabilistic graphical models (PGMs), inference and reasoning, automatic learning of PGMs, dynamic Bayesian networks, and machine learning in general. She has participated in some papers related to applications on distinct fields as environmental sciences, engineering, agriculture or medicine. Lately, she also works on image processing for scene localization, music genre classification and tasks related to robotics. She collaborates in research with colleagues from other Spanish and foreign universities in Europe (Denmark, France, and The Netherlands) and overseas (Australia and USA).