IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received January 27, 2020, accepted February 12, 2020, date of publication February 24, 2020, date of current version March 4, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2976033

A Multimodal End-to-End Deep Learning Architecture for Music Popularity Prediction

DAVID MARTÍN-GUTIÉRREZ, (Fellow, IEEE),
GUSTAVO HERNÁNDEZ PEÑALOZA, (Member, IEEE),
ALBERTO BELMONTE-HERNÁNDEZ, (Member, IEEE),
AND FEDERICO ÁLVAREZ GARCÍA, (Member, IEEE)
Visual Telecommunication Applications Group Signals, Systems and Radiocommunications Department, Universidad Politécnica de Madrid, 28040 Madrid, Spain

Corresponding author: David Martin-Gutiérrez (dmz@gatv.ssr.upm.es)

This work was supported by the H2020 European Project: EasyTv (https://easytvproject.eu/.) under Grant 761999.

ABSTRACT The continuous evolution of multimedia applications is fostering applied research in order to dynamically enhance the services provided by platforms such as *Spotify*, *Lastfm*, or Billboard. Thus, innovative methods for retrieving specific information from large volumes of data related with music arises as a potential challenge within the Music Information Retrieval (MIR) framework. Moreover, despite the existence of several musical-based datasets, there is still a lack of information to properly assess an accurate estimation of the impact or the popularity of a song within a platform. Furthermore, the aforementioned platforms measure the popularity in various manners, thus increasing the difficulties in performing generalized and comparable models. In this paper, the creation of *SpotGenTrack Popularity Dataset* (*SPD*) is presented as an alternative solution to existing datasets that will facilitate researchers when comparing and promoting their models. In addition, an innovative multimodal end-to-end Deep Learning architecture named as *HitMusicNet* is presented for predicting popularity in music recordings. Experiments conducted show that the proposed architecture outperforms previous studies in the State-of-the-Art by incorporating three main modalities to the analysis, such as audio, lyrics and meta-data as well as a preliminary compression stage via autoencoder to better the capability of the model when predicting the popularity.

INDEX TERMS Multimedia information retrieval systems, autoencoders, deep learning, feature compression, music information retrieval, popularity prediction, recommender systems.

I. INTRODUCTION

In recent years, the exponential growth of multimedia content regarding both size and diversity has caused the necessity of building powerful systems with the capability of managing large volumes of both structured and unstructured data [18], [19], [51]. Therefore, Multimedia Retrieval Systems (MRS) arise as a potential topic for both research and the development of effective applications that improve the data management process when searching for media objects in large-scale collections.

Moreover, many investigations are focused on solving the aforementioned challenges for different modalities including video [18], [21], [51], text [3] and audio [35], [50]. Most of the investigations mentioned above are based on performing the retrieval using a single modality. However, recent published research [64] is exploring multi-modality approaches that have been proved to be more efficient in Information Retrieval (IR) processes.

More specifically, an entire research branch in Music Information Retrieval (MIR) has recently received more attention due to the continuous evolution of the principal audio streaming platforms such as *Spotify*, *Lastfm* or *Billboard*. Thus, MIR applications play a crucial role to support these platforms with efficient and powerful algorithms and services such as recommender systems, music similarity approaches or detailed popularity analysis. Consequently, music providers may take advantage of these techniques to increase their profits by promoting their products and therefore, the quality of experience (QoE) of their customers.

Among the several applications that are currently being studied within the MIR framework, predicting the popularity

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

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see http://creativecommons.org/licenses/by/4.0/

of a song emerges as a potential multimedia application due to its huge impact and multiple benefits generated to all parties involved in the music industry. Firstly, artists and music producers may receive significant and objective information regarding the characteristics that achieve the highest popularity within the music market in a particular period of time. On the other hand, predictions can be used as a way of profiling listeners based on their tastes leading to the development of powerful recommender systems and even other commercial applications can take advantage of the benefits of this knowledge.

Additionally, the recent irruption of Machine/Deep Learning models (ML, DL) [21], [22], [56] has changed the paradigm in pattern recognition and classification tasks and are being used often within the MIR framework. These models have the universal capability of approximating any complex function by adapting its parameters according to the data available. In particular, DL models are a popular family of non-linear methods composed of multiple processing layers with the capability of learning complex representations of data. In addition, these techniques have enriched and improved the state-of-the-art in audio and image processing among other research fields. Thus, the availability of multiple modalities of information can be jointly used for better prediction/classification tasks. MIR comprises audio processing, lyrics analysis, ratings and authors meta-data among other tasks, therefore, DL models play an essential role to improve several MIR applications due to their competent performance when extracting automatically significant features and hidden patterns from raw data. Consequently, this paper has taken advantages of these native properties that DL has in order to implement a complete architecture that overcomes the state-of-the-art in predicting music popularity.

In this paper, two main contributions to improve previous solutions in music popularity prediction are presented:

i) A unified large-scale multimodal database to be used in diverse MIR applications such as popularity prediction, genre classification or music recommender systems.
ii) An end-to-end multimodal DL architecture named as *HitMusicNet* for music popularity prediction that has been demonstrated to be both more effective and precise than previous approaches.

Firstly, a multimodal updated database has been built as a fusion of two main platforms: *Spotify* and *Genius*, containing more than 100K tracks collected from 26 different countries, with the purpose of facilitating the research of similar topics. The following data sources are included:

i) a set of high-level audio features extracted directly from *Spotify*,
ii) a set of low-level audio features extracted from different audio representations such as Mel-spectrogram, *Tonnetz*, Chromagram or spectral centroids,
iii) a collection of text features directly gathered from lyrics,
iv) diverse information regarding artists such as the number of followers or his/her popularity,
v) the popularity of each track as a continuous value between 1 and 100 and the genres associated to each track,
vi) a complete set of previews of audio files as well as the full corpus of the available lyrics.

Furthermore, the proposed *HitMusicNet* architecture solves the popularity prediction issue from both regression and classification perspectives and outperforms the existing literature in this field, where most of the solutions are focused on ML techniques with limited capabilities to learn complex functions. To do so, both the proposed model and others suggested in previous studies were investigated using the proposed dataset in order to compare the results. Besides, these studies have mainly explored classification as a feasible solution to the problem by assuming that a track is popular if it exceeds a certain *popularity threshold*. Thus, this solution implies the definition of a subjective threshold and therefore, a hard-decision. Therefore, the proposed method provides an estimation of the popularity value and its corresponding error, leading to a soft-decision.

The remainder of this paper is organized as follows: in Section II, MIR related work is described. Section III outlines the main components involved in the proposed system. Section IV presents details of the text, audio and meta-data features. Afterwards, Section V describes the architecture of the proposed end-to-end architecture. Subsequently, in Section VI, the different experiments conducted and their results are described. Finally, the general conclusions and future work are outlined in Section VII.

II. BACKGROUND & RELATED RESEARCH

Recently, several authors have focused their research on mining musical track popularity information using ML techniques to incorporate an objective perspective to the Hit Song Science (HSS), which is of appealing interest to all the implicated stakeholders in the music industry. As described in [33], HSS is an MIR branch that addresses track popularity prediction to investigate whether a musical track will be a hit or not by applying ML techniques. The importance of this field lies in the opportunity of creating musical tracks based on those characteristics that, according to the most relevant experiments, have a larger probability of wide impact, and therefore, are potentially more attractive to the market. Although it is extensively known that both audio and lyrics are the principal characteristics that make a certain song popular, in diverse studies such as [6], [11] or [48], other factors related to the culture, the society or even the psychological perspective of audience, have also been taken into account.

Consequently, the proposed solution of this study is a complementary work to the most remarkable studies in this area described along the section. Firstly, in the experiments proposed in [11], a set of ML models are explored to classify songs as hit or not hit based on acoustic and lyrics information. Other authors [47], [48] focused their research on validating different state-of-the-art procedures that are capable of predicting the popularity of music titles from both global acoustic and human features. Nonetheless, the authors concluded that popularity cannot be measured and approximated by the use of these techniques.

In [49], a set of classifiers including Logistic Regression, Linear and Quadratic Discriminant Analysis and Support Vector Machines (SVM) were implemented to determine the popularity of a song as a binary classification problem. They employed *The Million track Dataset* [5] which provides a set of baseline features. Moreover, to improve their models precision, they incorporated additional features including a bag of words of a set of categorical features provided by this database. Finally, different regression models were presented to retrieve valuable information regarding the level of popularity. However, the main drawback of *The Million track Dataset* is that it only contains meta-data, neglecting raw audio files or lyrics which constrains the investigation of powerful features extracted from these sources.

In [32] and [33], an integrated data collection from different sources including *Lastfm* or *Spotify*, named as the Track Popularity Dataset (TPD) is proposed. This database contains around 20K tracks from 2004 to 2014 and it offers three feature-sets where the information from the Mel coefficients, the spectral centroids or the Chromagram representation are included. Moreover, they address two main topics: identifying a set of characteristics of music tracks to separate popular from non-popular tracks and the implementation of promising models with capabilities of learning from the profile from a popular track to predict the popularity of another.

In [38], a set of metrics and characteristics were proposed to improve the interpretation of how popularity can be measured considering the audio signal including both the complexity of the signal and a collection of Mel Frequency Cepstral Coefficients (MFCCs).

In [15], a set of experiments for HSS prediction is presented using data collected from Chinese and UK pop music charts. In these experiments, a set of audio features are selected to predict whether a song will be a hit or not. The authors remarked the existence of significant differences between the audio feature characteristics of Chinese hits and UK hits, and therefore, the systems should consider the diversity in musical cultures to be representative.

In a different scenario, [57], a highly efficient procedure was proposed for music analysis based on matrix similarity representations to predict the popularity of a certain song according to its similarity with others. In [30], a novel technique for audio representation is presented. In particular, authors proposed *DCAR*, a two-phased method that outperforms the state-of-art audio representations and has considerable benefits in event detection and audio-scene classification.

In [7], authors present three potential distance measures to determine the similarity among songs based on audio content including low/high level features as well as a hybrid combination.

In [58], the authors presented a hit detection model based on text features extracted directly from lyrics as the only source of information. To do so, they employed the Billboard Year-End Hot 100 singles dataset which contains singles from 2008 and 2013.

Moreover, in [46], an investigation to demonstrate how repetitive lyrics increase the probability of becoming a success in the market was published. In this work, the number one hits from Billboard, which contains songs from 1958 to 2012, is analysed in different controlled scenarios. The authors emphasize that repetitive lyrics are processed more fluently by consumers and therefore, they have more odds to triumph in the market. In addition, they conclude that this characteristic is potentially more influential than melodic repetition.

In [27], a model for detecting hit songs in dance music by using a database with tracks from 1985 to 2013 is proposed. They address the hit prediction framework as a classification problem using different features provided by The Echo Nest [14]. The study concludes that learning the representation of popularity is feasible using musical information.

In [37], authors explore the effectiveness of early stage popularity in order to predict what they call long-term popularity. For this purpose, they employ the Billboard Rock Songs Chart considering songs from 2010 to 2015. The proposed solution is formulated as a classification problem where the features are extracted from the audio signal.

Furthermore, authors in [45] have examined the correlation between the energy and the beats per minute and popularity in terms of sales. The principal objective of the study is to present the relationship between popularity and mood regarding the genre. They demonstrate a significant discrepancy in the mood score obtained for each genre.

The aforementioned investigations aimed to solve the popularity prediction problem using datasets of diverse size and features. The description of all the datasets can be found in [33], where authors seek to unify the information to make the performance of experiments easier.

Hence, several limitations were emphasized for existing research in MIR. To address and mitigate these constraints, this paper proposes the creation of *SpotGenTrack* Popular Dataset (*SPD*) to unify all the musical knowledge employing two of the most powerful platforms in musical and lyric content: *Spotify* and *Genius*. *SPD* contains not only several pre-calculated audio features but also the URL's of audio previews as well as the complete set of lyrics to avoid restrictions in new features extraction. In addition, distinct text features have also been included in the dataset as an initial approach to Natural Language Processing (NLP) that may contribute to future experiments. Finally, a set of socio-cultural features has been added as well to incorporate a different perspective when predicting music popularity.

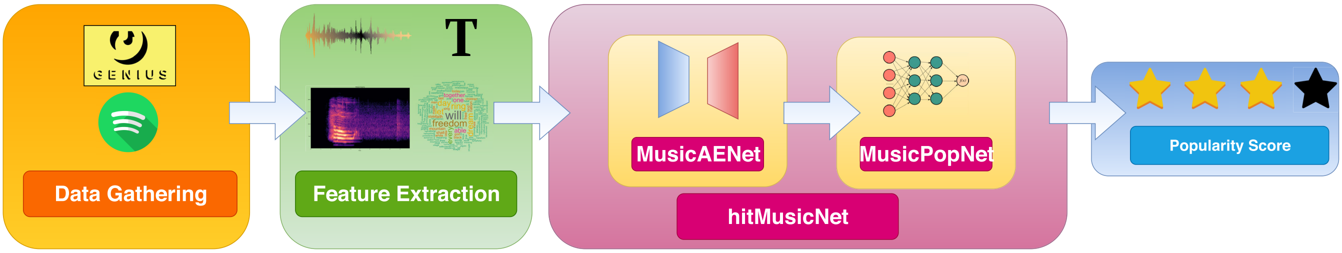

FIGURE 1. An illustrative High-level block diagram showing the pipeline of the presented study that attempts to promote the performance when prediction the popularity of a song.

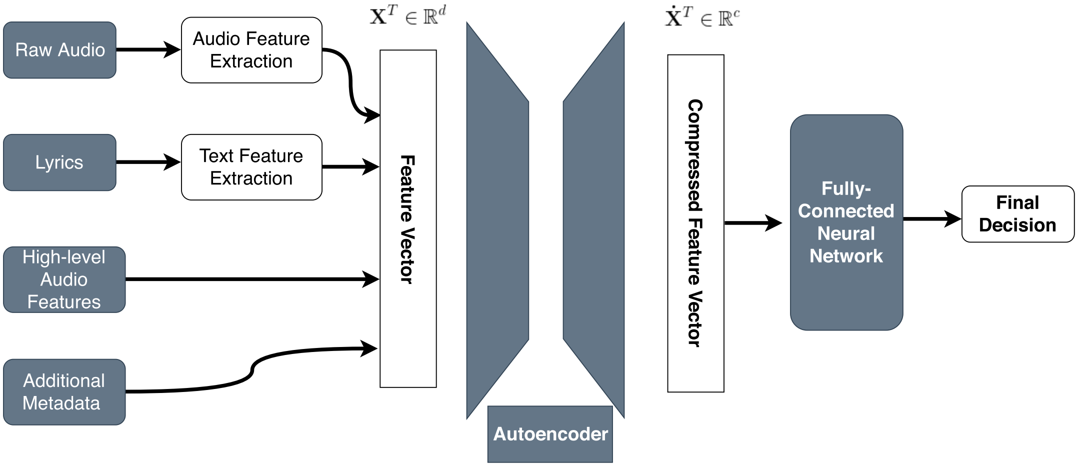
Raw Audio
Audio Feature Extraction
Lyrics
Text Feature Extraction
High-level Audio Features
Additional Metadata
$\mathbf{X}^T \in \mathbb{R}^d$
Feature Vector
Autoencoder
$\dot{\mathbf{X}}^T \in \mathbb{R}^c$
Compressed Feature Vector
Fully-Connected Neural Network
Final Decision

FIGURE 2. A general block schema outlining the principal functionalities and data components that form the proposed music popularity prediction system. After applying a feature extraction procedure for both the raw audio and the lyric components, the system obtains a high-dimensional feature vector and then, two additional steps are followed: a feature compression stage using an Autoencoder and a classifier via a fully-connected Neural Network.

Furthermore, an end-to-end DL architecture named as *MusicPopNet* is presented to solve the music popularity prediction from both regression and classification perspectives.

III. DATA ACQUISITION AND SYSTEM OVERVIEW

Along this section, the music popularity prediction problem is formulated from a mathematical point of view. Subsequently, the system architecture adopted to perform the experiments is deeply explained. Finally, the data gathering procedure to build *SPD* is analysed as an introduction to the relevant features extracted from it.

In Fig. 1, an illustrative description of the different stages that are considered in this paper to predict the popularity of a musical track are presented. In particular, three steps are required to manage it including: data gathering, feature extraction and finally, an end-to-end DL architecture named as *HitMusicNet* that provides the final popularity score.

On the other hand, Fig. 2 aims to stand out the main components that are employed to build the final system as well as to remark the different processes that each signal or data component follows throughout it. In particular, the system receives four data components as input including: i) a raw audio signal obtained via Spotify throughout the preview audio URL's, ii) the full corpus of the lyric obtained via Genius, iii) a set of high-level audio features provided directly from Spotify and iv) an additional set of meta-data based on the social information of the musical track. Additionally, both raw audio signals and lyrics pass through an intermediate step known as feature extraction to be represented in a more adequate manner. Subsequently, a solid feature vector is determined as the concatenation of the aforementioned representations of both audio and lyrics together with the last two data components. Such high-dimensional vector is then compressed using an Autoencoder named as *MusicAENet* and finally, a fully-connected layer named as *MusicPopNet* is employed to determine the final decision.

A. PROBLEM STATEMENT

One of the main goals of this paper consists in designing a robust architecture capable of predicting the popularity of a certain musical track given a wide set of features. Thus, the popularity of a track is modelled as a Random Variable (r.v.) $Y \in \mathbb{R}$, whereas $\mathbf{x}^T \in \mathbb{R}^d$ represents a collection of features extracted from different modalities. Moreover, the proposed DL architecture will provide an estimation of Y as a non-linear function of the input vector, so that, $\hat{Y} = h(\mathbf{x})$. Fig. 3 shows the distribution of the popularity of the promoted database *SPD*. It can be approximated as a Gaussian

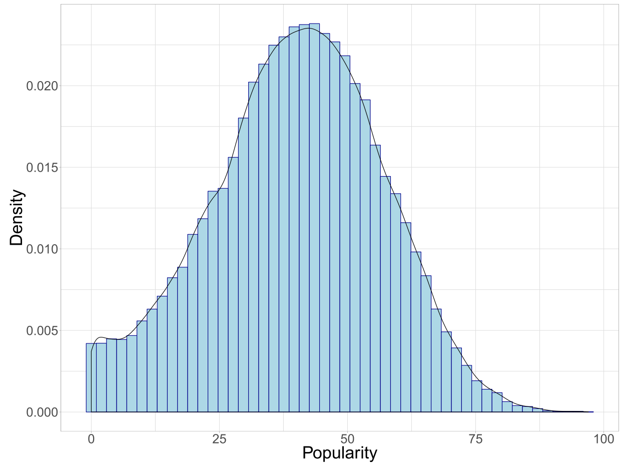

FIGURE 3. **Popularity distribution in the *SPD* Dataset with mean $\mu = 40.02$ and $\sigma = 16.79$.**

distribution with parameters $\mu = 40.02$ and $\sigma = 16.79$. Therefore, it is clear that taking part of the top list is very complicated according to the distribution and only a limited set of musical tracks achieve this objective.

Moreover, $\mathbf{x}^T \in \mathbb{R}^d$ is a high-dimensional feature vector which contains information from the aforementioned modalities. However, working with high-dimensional feature vector is not always adequate due to the high probability of suffering from overfitting [9]. Therefore, an Autoencoder [24] is used to compress the representation of such feature vector in a lower-dimensional space $\mathbb{R}^c$ where $c \ll d$. We denote $\dot{\mathbf{x}}^T$ as the compressed representation of the feature vector that will be used to obtain an estimate of the popularity, $\hat{Y}$. In the final stage, $\dot{\mathbf{x}}^T$ feeds a fully-connected DL model which returns the final level of popularity.

B. SPOTGENTRACK *Popularity Dataset* (SPD)

The *SpotGenTrack* (*SPD*) is a collection of information devoted to being used in multiple MIR disciplines such as genre classification, system recommends or auto-tagging. In particular, this paper uses the dataset to predict popularity in musical tracks.

The data gathering procedure was carried out via *Spotify*[1] and *Genius*[2] API's. To reinforce the popularity representation around the world, more than 26 countries where *Spotify* is available, have been analysed during the data gathering phase. More specifically, for each country, the top 50 playlists per category were collected. On the other hand, all the lyrics were extracted throughout the API of *Genius*.

As a result, the proposed *SPD* dataset is composed by a total of **101.939 Tracks**, **56.129 Artists** and **75.511 Albums**, and consequently, it contains a precise representation of the popularity among the different countries and genres.

IV. MULTIMODAL FEATURE EXTRACTION

In this section, a collection of features evoked from the different modalities including text, audio and meta-data, are described to emphasize the necessity of their incorporation into a unique feature vector $\mathbf{x}^T$. These features lead to improve the performance of the proposed end-to-end DL architecture described in Section V.

A. TEXT FEATURES

Firstly, a set of descriptors is extracted regarding the corpus of the lyrics. The reason of using this modality lies in improving the final system by combining text features together with audio descriptors. Thus, using NLP techniques, a *stylometric* analysis is performed and the following features are obtained: i) The total number of sentences, ii) the average number of words per sentence, iii) the total number of words, iv) the average number of syllables per word, v) a sentence similarity coefficient and vi) a vocabulary wealth coefficient.

Furthermore, both the sentence similarity and vocabulary wealth coefficients require an additional processing that will be presented in the following sections.

1) SENTENCE SIMILARITY COEFFICIENT

The sentence similarity coefficient κ has been determined to investigate the influence and the correlation of repetitive patterns in lyrics with the popularity of the musical track.

There exist several approaches to address this problem as mentioned in [1], [34], [42]. However, in this paper the sentence similarity coefficient is computed based on the Term frequency-Inverse document frequency *Tf-Idf* [53] and the computation of the cosine distance as Algorithm 1 describes. In particular, the objective of the algorithm is to define a similarity matrix based on the cosine distance metric *Tf-Idf*. Subsequently, the method remains only the upper diagonal elements of the resulted symmetric matrix which are larger than a similarity threshold μ and discards the rest. Empirical experiments performed allowed us to set this threshold to 0.75. Moreover, $\kappa \in [0, 1]$, where $\kappa = 1$ means that all the sentences of a given lyric are the same, whereas $\kappa = 0$ indicates the contrary statement.

Algorithm 1 Sentence Similarity Coefficient

```
procedure SENTENCESIMILARITY(lyric)
    if |lyric| > 1 then
        $l \leftarrow lyric$
        $tf\_idf \leftarrow computeTfIdf(l)$
        $m \leftarrow computeCosineSimilarity(tf\_idf)$
        $diag \leftarrow getUpperDiagonal(m)$
        $t \leftarrow |diag|$
        $n \leftarrow 0$
        for s in diag do
            if s $\geq \mu$ then
                $n \leftarrow n + 1$
        $sim \leftarrow n/t$
    return $sim$
```

[1] Spotify Developer. Available at: https://developer.spotify.com/
[2] Genius Developer. Available at: https://genius.com/developers

2) VOCABULARY WEALTH COEFFICIENT

The vocabulary Wealth Coefficient ρ is and indicator of the diversity of the vocabulary employed when writing a certain lyric. Firstly, we define the set of non-stop-words from a corpus as $W = \{w_i\} \ \forall \ i = 1, \ldots, r$. Then, a set of **distinct words** is calculated to retrieve the length of vocabulary that a given corpus has and it is defined as the vocabulary of the corpus and denoted as $W_d = \left\{w_j\right\} \ \forall \ j = 1, \ldots, n$, where $r \geq n$. Subsequently, the frequency distribution of W_d is calculated and sorted to emphasize those words with more likelihood regarding the frequency distribution. Additionally, the words from W_d which provide the 85% of the cumulative distribution are gathered in a vector new vector W_c.

Finally, ρ is calculated as a ratio of the length corresponding to the non-repetitive words of W_c and the total number of words in the lyric. Therefore, $\rho = |W_c|/|W|$, where $\rho \in [0, 1]$ refers to the vocabulary diversity (wealth) metric and $|W_d|$, $|W|$ are the lengths of the sets W_d and W respectively.

B. HIGH-LEVEL AUDIO FEATURES

Spotify provides a significant set of high-level audio features that can be easily interpreted by professionals and researchers. A complete description of these descriptors is depicted in Table 1. Among the available features, the popularity of a song is observed, which is used as the label or response variable during the training process of the end-to-end architecture.

TABLE 1. High-level audio features description provided by ***Spotify***.

Audio Features	Description
Acousticness	A confidence measure $\in [0, 1]$ of whether the track is acoustic.
Danceability	A value $\in [0, 1]$ that describes how suitable a track is for dancing based on a combination of musical elements including tempo, rhythm stability, beat strength, and overall regularity..
Duration	The duration of the track in ms
Energy	A confidence measure $\in [0, 1]$ that represents a perceptual measure of intensity and activity.
Instrumentalness	A measure $\in [0, 1]$ that predicts whether the track contains no vocals.
Key	The key the track is in. Integers map to pitches using standard Pitch Class notation $\in \{0, 1, 2, \ldots 11\}$.
Liveness	A value $\in [0, 1]$ of whether there is presence or not of an audience in the recording and indicates a likelihood.
Loudness	The overall loudness of a track in decibels (dB).
Mode	Indicates the modality (major or minor) of a track. Major is represented by a 1 and minor by a 0.
Speechiness	A measure $\in [0, 1]$ of whether the presence of spoken words in the track is or not detected.
Tempo	The overall estimated tempo of a track in beats per minute (BPM).
Valance	A measure $\in [0, 1]$ describing the musical positiveness conveyed by a track where high valence indicate sound positive and lower valence indicates sound more negative.
Popularity	A measure $\in [0, 100]$ describing how popular a track is, where 0 means no popular and 100 very popular.

C. LOW-LEVEL AUDIO FEATURES

This section describes the multiple low-level audio features extracted after processing the 30 seconds audio previews stored in the proposed *SPD* dataset. This set of features includes energy descriptors as well as both timbre and harmonics descriptors.

The following standard parameters in audio processing were used to compute all the features:

i) the sampling rate denoted as *(sr)* was fixed to 44100, ii) the analysis window length (W_L) was set up to 30 seconds, iii) the number of overlapped samples or Hop length (H_L) was fixed to 512 samples and iv) a total number of 2584 audio frames for each representation which is obtained using (1).

$$a_f = \frac{W_L \ sr}{H_L} \tag{1}$$

Moreover, a collection of statistics including first and second moments are computed for each feature by considering all the audio frames in order to have a solid low-level audio representation.

1) MEL SPECTROGRAM AND MEL FREQUENCY CEPSTRAL COEFFICIENTS (MFCCS)

The computation of the MFCCs for modelling music signals was firstly investigated in [17]. This set of coefficients is included to improve the audio signal representation.

Hence, the Mel-scaled spectrogram is calculated as [40], [54], [63] describe. This spectrum consists in a time-frequency representation of the sound where the powers of the spectrum obtained in the Fourier transform are mapped into a number of points equally spaced in time and frequencies regarding the Mel scale. The number of coefficients required to represent the sound signal properly depends on the final application as well as the capabilities of the system. In this paper 40 MFCCs are calculated.

Furthermore, the Mel-spectrogram, denoted as *MEL*, is computed as well setting up the following parameters: i) The length of the Fast Fourier Transform was fixed to 2048 samples, ii) the number of filter bands selected was 127, iii) The frequency range of the analysis lies in the interval ($50Hz$, $22500Hz$) and iv) a *Hamming* window [60] was used to perform the spectral analysis.

The Mel-scaled representations are typically used to capture timbral aspects of music. However, they are not meaningful regarding the pitches. Indeed, the lack of information concerning the harmony encoding of a musical track is an

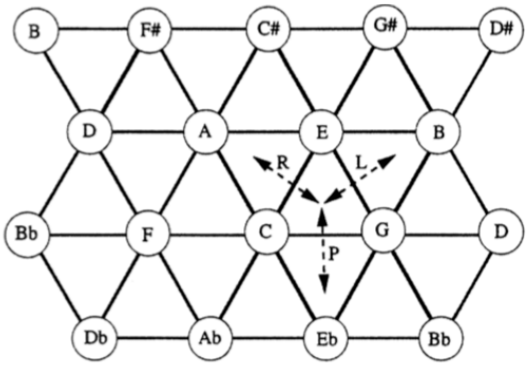

FIGURE 4. **Representation of the Harmonic Network or *Tonnetz* in a grid format where the circles consist in the set of keys whereas the triangles are the formed chords. Letters *R*, *L* and *P* corresponds to Relative, Leading tone and Parallel, the so-called PRL operations.**

essential reason to investigate both *Tonnetz* and Chromagram representations.

2) THE HARMONIC NETWORK OR *TONNETZ*

The Harmonic Network or *Tonnetz*, was proposed in [25] as a novel approach for detecting changes in the harmonic content of musical audio signals. This procedure is a planar representation of pitch relations where close harmonic relations are modelled by small distances on the plane. Moreover, lines of fifths go from left to right whereas major thirds travel from bottom left to top right and minor thirds from top right to bottom left. In Fig. 4, an example of the *Tonnetz* grid is depicted, where circles represent musical keys and triangles form different chords. The grid is created by unrolling the circle of fifths horizontally, staggered vertically to place the major third and sixth above and minor thirds and sixths below each tone as it is explained in [4]. In addition, P, R and L correspond to Parallel, Relative and Leading tone and are known as the neo-Riemannian *PRL*-group operations described by Richard Cohn in [10].

Since the *Tonnetz* estimates tonal centroids as coordinates of a six-dimensional interval space as it is explained in [25], Υ consists in a six-dimension feature representation at each audio frame.

3) CONSTANT-Q CHROMAGRAM

As reviewed in [2], the conventional linear frequency representation provided by the Discrete Fourier Transform gives a constant separation between coefficients for musical sounds consisting of harmonic components. On the other hand, the log-frequency representation provides a constant pattern of the spectral components and therefore, identifying a certain instrument or the fundamental frequency is a straightforward problem.

A *Chromagram* representation [55] consists of a twelve-element vector indicating how much energy is released by each pitch class. We will denote the Chromagram as Ψ and $\varphi = \{C, C\#, D, D\#, E, F, F\#, G, G\#, A, A\#, B\}$ as the set of possible pitches. Hence, feature Ψ is a 12-dimension representing the harmony of the song.

4) OCTAVE-BASED SPECTRAL CONTRAST

The *Octave-based Spectral Contrast* representation [29], defined as Φ, considers the spectral peak, the spectral valley and their difference in each sub-band. In several musical tracks, the strongest spectral peaks usually come from the harmonic components whereas the non-harmonic components are more often to appear at spectral valleys.

Therefore, Φ reflects the relative distribution of the harmonic and non-harmonic components in the spectrum [29]. Consequently, by adding Φ to $\mathbf{x}^T$, the relative spectral information is preserved.

Some of the principal distinctions between the MFCCs and the Spectral Contrast procedure lies in the selection of the filter bank. The former uses the Mel-scale filter whereas the latter employs an octave-scale filters. In addition, the former sums the Fast Fourier Transform (FFT) amplitudes at each sub-band whereas the latter extracts the spectral peaks, valleys and their differences. Finally, in the final stage of the procedure, the former method calculates the Discrete Cosine Transform (DCT) whereas the latter uses the Karthusen-Loeve transform [13]. Thus, Φ is a 7-dimensional representation.

5) SPECTRAL CENTROID

The *Spectral Centroid* [61], represented by ζ, is an indicator of the frequency where the energy of the spectrum is centred upon. In particular, it is computed as an average of the frequency weighted by the spectral magnitude as it is defined in (2):

$$\zeta = \frac{\sum_k S(k)f(k)}{\sum_k S(k)}, \tag{2}$$

where $S(k)$ refers to the spectral magnitude at frequency bin k and $f(k)$ indicates the frequency at bin k.

6) SPECTRAL BANDWIDTH

The *Spectral Bandwidth* computes the p-order spectral bandwidth as it is defined in (3):

$$SB = \left(\sum_{k=1}^{K} S(k)(f(k) - \zeta)^p\right)^{\frac{1}{p}} \tag{3}$$

where $S(k)$ refers to the spectral magnitude at frequency bin k, $f(k)$ indicates the frequency at bin k and ζ refers to the aforementioned *spectral centroid*.

7) ZERO-CROSSING RATE

As it is described in previous studies including [41], the so-called *Zero Crossing Rate* (ZCR) is a common descriptor employed in speech recognition, audio classification and segmentation and many other MIR topics, so that, it was included in these experiments as well to promote the low-level audio representation.

More specifically, ZCR is defined as the rate of sign changes detected along a digital signal, so that, we are measuring the rate whether the signal has changed from positive to negative or vice versa. Additionally, it is well-known that periodic sounds yield to a smaller ZCR whereas noisy sounds tend to have a larger value.

D. ADDITIONAL META-DATA FEATURES

One of the objectives of the paper consists of representing a song as a low-dimensional feature vector based on multiple modalities. In previous sections, both audio and text representations were computed. However, when predicting the popularity of a song, additional information related to both social framework and marketing strategies must be considered as well in the analysis. Therefore, a set of meta-data parameters are also considered in the final representation of the song including: i) the number of followers of the artist, ii) the popularity of the artist and iii) the number of available markets where the song will be or is released.

E. FINAL FEATURE VECTOR

Finally, a feature vector, denoted as $\mathbf{x}^T$ in Section III-A, is obtained as a result of concatenating the aforementioned multimodal features.

Subsequently, $\mathbf{x}^T$, is passed through the proposed end-to-end architecture which will firstly both compress and remove redundancy of the representation via Autoencoders and finally, predict the value of the popularity of the input song.

V. *HITMUSICNET*: AN END-TO-END DL ARCHITECTURE

This section is devoted to describing the proposed *HitMusicNet* architecture for predicting the popularity of a song based on the feature vector obtained in Section IV-E. The architecture is divided in two main steps: 1) a deep autoencoder [24], [62] named as *MusicAENet* and 2) a deep neural network named as *MusicPopNet*.

The former performs a feature compression transformation by encoding possible correlations and redundancy as well as reducing noise from the input data. This additional step is crucial to avoid the so-called overfitting problem [9]. The latter receives the compressed representation and predicts the value of its popularity.

A. FEATURE COMPRESSION VIA MUSICAENET

The general architecture of an Autoencoder (AE) consists in two different stages: an encoder and a decoder. In our proposal, several experiments were conducted, and the best configuration consisted of two hidden layers with $d/2$ and $d/3$ neurons respectively for both encoder and decoder, where d indicates the dimension of the input data. Moreover, the level of compression in the encoding layer is denoted as δ and indicates the number of neurons in such layer. More specifically, it was empirically investigated using different values including $\{1/4, 1/5, 1/7\}$, leading to $1/5$ as the most appropriate compress level. Additionally, some preliminary experiments were performed without any compression, but the performance was not as accurate as when a small level of compression is added.

Moreover, the Mean Squared Error (MSE) loss function was employed to both training and validating the AE. Additionally, the Adam optimizer algorithm [36] was selected to determine the stochastic gradient descent using a learning rate equal to 0.001. During both training/validation phases, the loss function decayed around 10^{-5} indicating the negligible loss of information for such level of compression.

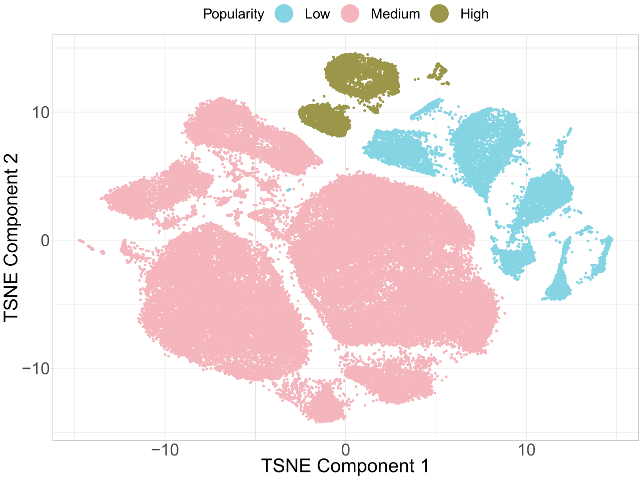

FIGURE 5. TSNE representation of the compressed feature $\dot{\mathbf{x}}^T$ regarding the popularity level: Low (Orange), Medium (Purple) and High (Green). This procedure was obtained for a the medium compression level $\delta = 1/5$.

In Fig. 5, a 2D-embedding representation was obtained after applying the *t-distributed Stochastic Neighbor Embedding* (TSNE) [43] algorithm to $\dot{\mathbf{x}}^T$ in order to visualize its components regarding the popularity level. Hence, employing $\dot{\mathbf{x}}^T$ a potential representation of the data is assessed to apply either classification or regression techniques to solve the musical popularity prediction problem. More specifically, the TSNE was performed by grouping the data points into three popularity classes (Low, Medium and High). The discrete representation of these three classes regarding the music popularity is denoted as r.v. Y following the expression depicted at (4).

$$Y = \begin{cases} 0, & 0 \leq p < 25 \\ 1, & 25 \leq p < 65 \\ 2, & 65 \leq p \leq 100 \end{cases} \tag{4}$$

where p is the popularity parameter commented in Section IV-B. In particular, $y = 0$ corresponds to Low popularity whereas $y = 2$ refers to High popularity. This representation will be used in Section VI when predicting popularity from a classification perspective.

B. PREDICTING POPULARITY VIA MUSICPOPNET

The proposed architecture of *MusicPopNet* is depicted in Fig. 6. The arrows represented on it indicate that all the

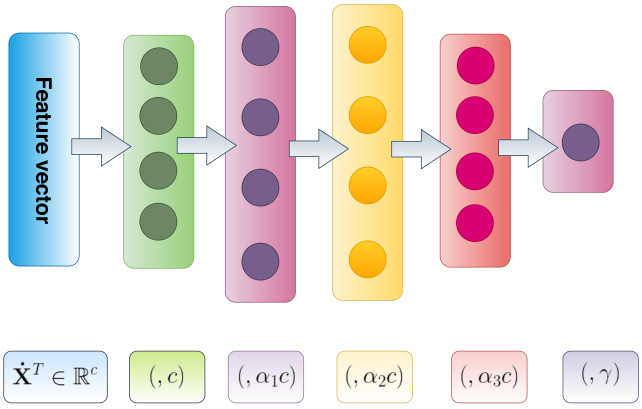

FIGURE 6. **Illustrative schema of the proposed *MusicPopNet*. The arrows in the figure indicates that the all the layers are fully connected to each other.**

layers are fully-connected. The input of this network is the output of the aforementioned AE model *MusicAENet*. On the other hand, the output of the network is the popularity level.

The architecture presented in Fig. 6, comprises fully-connected layers with a particular number of neurons based on three parameters α_1, α_2 and α_3. These parameters have been empirically investigated for distinct models to determine the appropriate number of neurons in the hidden layers. There is no an exact method to solve this problem, but many authors suggest different approaches based on rule-of-thumb as presented in [31].

Firstly, in a DL model with L layers, a subset of $L-2$ layers forms the hidden layers. Moreover, in this subset, an adequate selection of the number of neurons or units is crucial to increase the capabilities of the system to learn patterns. Furthermore, the hidden layer set has been partitioned into three categories depending on its position in the global architecture of the model: i) *Initial Hidden Layer*, ii) *Intermediate Hidden Layer* and iii) *Final Hidden Layer.* Each of the above categories is composed by a different number of neurons $\lambda_i = \lfloor \alpha_i c \rfloor \forall i = \{1, 2, \dots L-2\}$, where c represents the dimension of the input $\dot{\mathbf{x}}^T$. Therefore, these parameters are scalars that control the number of neurons per hidden layer. After performing different experiments using the criteria described in [31], the set of parameters $\{\alpha_1, \alpha_2, \alpha_3\}$ was determined as $\{1, 1/2, 1/3\}$ respectively. Hence, the dimension of the hidden layers decreases linearly with respect to the input dimension.

Furthermore, the selection of an adequate optimizer is a critical factor in the training performance. Consequently, the principal well-known optimizers have been considered in this part of the study including Stochastic Gradient Descent algorithm *SGD* [8], *Adam* optimizer proposed in [36], *Adadelta* method [65] and a derivation of the Adam procedure named as *Nadam* [12].

Another essential parameter to be selected is related to the initialization of the set of weights for each of the neurons. There are several methods that have been investigated in the literature with the aim of optimizing the learning process and reducing the vanishing gradient problem [28]. In this paper, two principal families of weight initialization procedures have been analysed along the experiments including: i) *Xavier normal and uniform initializer* [20], ii) *He normal and uniform initializer* [26]

Moreover, the loss function selected to be minimized in this case is the classical Minimum Square Error (MSE).

Furthermore, the activation function applied by all the hidden layers arises as a potential parameter to be established. Along these experiments, all the hidden layers use *ReLU* (Rectifier Linear Unit) function [39] as the non-linear activation function due to its ability of speeding up the training process. More specifically, the output layer is different regarding the sort of problem to be addressed. In Fig. 6, the dimension of the output layer is denoted with γ. Hence, from a regression perspective, it is composed by a unique neuron and a Sigmoid activation function [23] with the aim of getting the final prediction in the range [0, 1]. Hence, since the original value of the popularity lies in the interval [0, 100], a normalization procedure is needed to adapt the range to the one required by the neural network. On the other hand, from a classification perspective, $\gamma = 3$, which is the number of popularity categories. In this case, the activation function is the Softmax function [44].

Additionally, all the hidden layers have a regularizer parameter to avoid suffering from overfitting. There exist several techniques but the Dropout [59] one was selected in this study since it speeds up the training process while maintaining the precision regarding performance. More specifically, this method randomly drops a set of units from the neural network along the training process to prevent the units to co-adapt too much. In our experiments, the parameter of drop proportion is represented as ρ and different standard values $\{0.25, 0.5, 0.75\}$ were investigated when conducting the experiments.

C. TRAINING AND VALIDATION PROCEDURE

Firstly, the data is scaled and divided into the two well-known sets: Training and Testing, with 80% and 20% of the data points respectively. The former set is used to train and validate the model using a Cross-Validation procedure [52], whereas the latter is employed to analyse the capability of the model to generalize whether new data is introduced. In particular, a Stratified Cross-Validation (SCV) method [66] is employed to promote the quality of the estimation by providing balanced intra-class distributions when partitioning the dataset in k-folds. More specifically, several experiments were carried out using $k = \{5, 10\}$ to investigate the influence of k in the performance of the model in terms of both CPU time and quality in the performance.

Moreover, the training procedure was improved via Early Stopping method [9], which avoids the model to overfit by stopping the training process when the validation error does not decrease after ψ steps, where ψ has been fixed to 10 epochs and the total number of epochs was fixed to 100.

TABLE 2. Comparison between the three best models obtained during the experimental stage in terms of the features selected in the training process. α, β and γ are a set of parameters used to control the number of hidden neurons, Layers refers to the number of hidden layers and δ indicates the level of compression respect to the original input vector.

Model	*Config*	*Layers*	α_1	α_2	α_3	Optimizer	Weight initializer	δ	CPU (ms)
1	A	3	1	1/2	1/3	Adadelta	Glorot Uniform	1/5	135k
1	B	3	1	1/2	1/3	Adam	Glorot Uniform	1/5	134k
1	C	3	1	1/2	1/3	Adadelta	He Normal	1/5	113k
1	D	3	1	1/2	1/3	Adam	He Normal	1/5	118k
2	A	3	1	1/2	1/3	Adadelta	Glorot Uniform	1/4	110k
2	B	3	1	1/2	1/3	Adam	Glorot Uniform	1/4	113k
2	C	3	1	1/2	1/3	Adadelta	He Normal	1/4	125k
2	D	3	1	1/2	1/3	Adam	He Normal	1/4	161k
3	A	3	1	1/2	1/3	Adadelta	Glorot Uniform	1/7	108k
3	B	3	1	1/2	1/3	Adam	Glorot Uniform	1/7	114k
3	C	3	1	1/2	1/3	Adadelta	He Normal	1/7	111k
3	D	3	1	1/2	1/3	Adam	He Normal	1/7	150k

Finally, in order to compute the final evaluation metrics, the mean for each metric over all the *SCV* iterations is computed.

VI. EXPERIMENTAL RESULTS

In this section, the experiments conducted using the *SPD* dataset are presented. *SPD* has a similar popularity distribution in comparison to *The One Million Song Dataset* (MSD) [5] created in 2011, as well as the Track Popularity Dataset (TPD) created in 2014 [32]. This factor is essential in order to compare the presented results with previous investigations.

As mentioned in Section II, many authors formulated the music popularity prediction as a binary classification by defining a threshold that separates Hit/Non-hit musical tracks. However, a decision-making procedure is required to establish such threshold which may lead to imprecise solutions. Additionally, as described in [49], a classification approach, specially a binary one, is a simplification of the problem where valuable information is missing. However, a set of experiments has been conducted from a classification perspective to show the performance of the proposed architecture in both scenarios.

During the experiments, vector $\dot{\mathbf{x}}^T$ was employed using different values of the compression level parameter δ introduced in Section V-A. Therefore, the goal of these experiments consists of analysing different parameters of both *MusicAENet* and *MusicPopNet* to achieve the best performance of the end-to-end architecture.

The identification of the proposed models is the following: Model Z - Ξ, where $Z = \{1, 2, \ldots, m\}$ and indicates the number of models designed. On the other hand, $\Xi = \{A, B, C, D\}$ represents the four configurations regarding the neural network parameters described in Section V.

A. REGRESSION

From a Regression point of view, different models have been implemented and evaluated based on the architecture and the parameters explained in Section V-A and V-B. In Table 2, a summary of the best models regarding their performance when predicting the popularity is presented. More specifically, the four configurations of each model are presented including the level of compression in *MusicAENet* (denoted by δ) with respect to the original feature vector as well as the computational time in milliseconds. In addition, the batch size was fixed to 256, the number of hidden layers was fixed to 3 and the number of k-folds was 5 in all of the models described in Table 2.

Furthermore, the results obtained during the *SCV* process in terms of Mean Absolute Error (MAE) and Mean Squared Error (MSE) are summarized in Table 3. Those configurations that assessed the best results for each model are remarked in bold. As it is observed, even when the level of compression in *MusicAENet* is very high ($\delta = 1/7$), the MAE still remains around 0.9 in the three stages: training, validation and testing.

1) DISCUSSION

As Table 2 shows, all the models have the same architecture in terms of number of hidden layers and number of neurons per hidden layer which are controlled by parameters α_1, α_2 and α_3. However, both the weights initializer and the optimizer are crucial to assess the highest quality performance and avoid an expensive computational cost in terms of CPU (ms).

Furthermore, the results of the experimental models in terms of MSE and MAE are presented in Table 3, where the bold rows indicate the best configuration for each model regarding the MAE and the CPU in the three sets: train, validation and test. As expected, if the compression level is too high (small δ), the resulted model cannot estimate the popularity as accurate as if the compression level is more adequate ($\delta = 1/5$). On the other hand, when the compression is too small, the number of features is higher and some of them may contaminate the estimation with noise.

In terms of computation cost, it is clearly observed that the higher the level of compression, the less amount of resources is needed to be trained. Thus, depending on the final application and the technical resources, a configuration that does not achieve an optimal solution in terms of MAE can be selected,

TABLE 3. Regression metrics belonging to the DL models. The corresponding configurations of such models are presented in Table 2. The bold rows indicate the best performance in terms of MAE.

Model	*Config*	*MSE Training*	MSE Validation	*MAE Training*	MAE Validation	MAE Test
1	**A**	**0.0120**	**0.0112**	**0.0868**	**0.0844**	**0.0855**
1	B	0.0113	0.0117	0.0845	0.0857	0.0859
1	C	0.0119	0.0122	0.0867	0.0886	0.0870
1	D	0.0115	0.0127	0.0848	0.0906	0.0911
2	A	0.0115	0.0132	0.0851	0.0926	0.0919
2	**B**	**0.0113**	**0.0120**	**0.0844**	**0.0875**	**0.0876**
2	C	0.0117	0.0136	0.0859	0.0933	0.0941
2	D	0.0111	0.0124	0.0835	0.0898	0.0913
3	A	0.0124	0.0135	0.0881	0.0937	0.0924
3	B	0.0120	0.0131	0.0867	0.0908	0.0912
3	**C**	**0.0123**	**0.0121**	**0.0879**	**0.0870**	**0.0884**
3	D	0.0117	0.0122	0.0855	0.0885	0.0901

TABLE 4. Classification metrics including accuracy, precision, recall and f1-score obtained during the validation phase of *SCV* strategy regarding the level of compression. See Table 2 for more specifications of the models.

Config.	*Cat. Cross-Entropy*	*Accuracy(%)*	*Recall (%)*	*Precision (%)*	*F1-Score (%)*	*CPU (ms)*
1-A	0.4128	83.03	83.01	83.04	83.02	136k
2-B	0.4086	83.46	83.46	83.47	83.47	158k
3-C	0.4218	82.65	82.64	82.66	82.65	131k

leading to a considerable reduction in terms of computational cost.

Hence, **model 1-A** was selected as the best model since it provides an appropriate trade-off between having a good performance (avoid overfitting) and a low computational cost.

Finally, Table 5 compares the performance of several existing works in the same field with the results of the proposed end-to-end architecture. From a regression perspective, it, outperforms previous studies. Nevertheless, a limitation lies in the comparison due to the diversity of datasets employed for different authors. Consequently, this paper includes the construction of *SPD* dataset that can be used for future studies using the same information and criteria.

To mitigate this problem, the experiments described in previous studies such as [49], have been reproduced using some of the suggested models over *SPD* instead of *MSD* to analyze relevant differences. As Table 5 presents, the proposed *MusicPopNet* decreases in terms of MAE in the testing set in comparison with the rest of the models and reaches a smaller level of convergence in terms of MSE. Moreover, a comparison with [33] from a regression perspective cannot be provided properly due to the different characteristic of the suggested models that works with time series variables.

B. CLASSIFICATION

In this section, the problem is faced from a multi-class classification perspective where in this case, Y is a discrete r.v. with a set $\mathcal{C}$ of feasible values: $\{0, 1, 2\}$ as (4) suggests. These discrete values can be interpreted as {"Low", "Medium", "High"} popularity. Therefore, since originally Y is a continuous r.v., a transformation procedure is required to represent Y as a discrete r.v.. To do so, two thresholds, ϵ_0 and ϵ_1, are established based on the quartiles of the distribution of Y. Consequently, $\epsilon_0 = Q_1$, where Q_1 indicates the first quartile, and $\epsilon_1 = Q_3$, being Q_3 the third quartile of the distribution of Y.

Moreover, the classification task is performed by applying the same architecture of *MusicPopNet* presented in Section V-B with a minor change in the output layer of the network. More specifically, since there are three different classes, the output layer needs to have 3 neurons and a *Softmax* as its activation function.

The objective of this scenario is the same as in the previous one: to find the best architecture for the two components of the end-to-end architecture. To limit the amount of experiments performed in this scenario, the training/Validation/Testing procedure is repeated for the three model configurations that achieved the best performance in the regression scenario including: model *1-A*, model *2-B* and model *3-C* (See Tables 2 and 3 for more details). In Table 4, the results obtained by these three models are presented from a classification perspective. To measure the quality of the model, the standard classification metrics are used to evaluate them including accuracy, precision, recall and f1-score [16]. Additionally, the loss function that is minimized when training the model is the so-called Categorical Cross-Entropy [67].

Finally, Table 4 shows the classification metrics that are used to evaluate the performance of the models. More specifically, the metrics were computed by taking the mean value over the different iterations of the *SCV* method. Although all the presented models attained good results, model *1-A* offers the best trade-off between computational cost and accuracy and it was selected to be the final configuration for the component denoted as *MuscPopNet*.

TABLE 5. Comparison among the existing models and the proposed approach named as *MusicPopNet* (model 1-A) from a regression perspective. See Table 2 for more specifications regarding the model parameters.

Name	*Study*	Dataset	ML Problem	*MSE* Testing	*MAE* Testing
MLR	[49]	MSD	Regression	0.0184	0.1357
MLR + Lasso	[49]	MSD	Regression	0.0180	0.1342
MLR	Proposed	*SPD*	Regression	0.0163	0.1034
MLR + Lasso	Proposed	*SPD*	Regression	0.0145	0.0986
HitMusicNet	**Proposed**	***SPD***	**Regression**	**0.0118**	**0.0855**

TABLE 6. Comparison among the existing models and the proposed approach in terms of Classification. Regarding the set of acronyms from the datasets: MSD (Million Song Dataset), *SPD* (SpotGenTrack Popularity Dataset), TPD (Track Popularity Dataset). The proposed DL model is identified as 1-A indicating Model 1 and configuration A.

Model-identifier	*Research*	Dataset	Classes	Precision (%)	Recall (%)	F1-score (%)
LDA	[49]	MSD	2	61.01	52.90	56.70
LDA	Proposed	*SPD*	2	64.10	60.40	62.20
MLP	[49]	MSD	2	58.80	44.10	50.40
MLP	Proposed	*SPD*	2	70.90	79.20	74.80
Rhythm+PCA (100)+MFCC	[33]	TPD	2	39.40	32.10	35.40
Hit's range (1-50)	[6]	Billboard	2	75.80	75.30	75.10
HitMusicNet	**Proposed**	***SPD***	**3**	**83.04**	**83.01**	**83.02**

1) DISCUSSION

According to the results presented in Table 4, in order to have the best trade-off between quality in the performance and computational cost, we selected model *1-A* in both classification and regression case. Consequently, this model is proposed as the *MusicPopNet* model within the whole end-to-end architecture.

Furthermore, unlike other proposals, this approach has increased the number of classes of popularity up the three with the aim of adding an intermediate level of popularity so that, hard and binary decisions are relaxed. In addition, following the same criteria described in the regression scenario, a set of models proposed in previous studies has been reproduced to compare their performance in our dataset *SPD* together with the proposed end-to-end architecture achievement. The results are summarized in Table 6, where it is shown that *HitMusicNet* reaches a better performance regarding the classification metrics even increasing the number of classes.

All the experiments were performed using a computer with an Intel I7 9th generation CPU, 16 GB RAM and a NVIDIA GTX 970 GPU. The entire dataset containing the tracks, artists and albums information is now available at **Mendeley** for both testing purposes and comparison with the work described here.

C. FINAL END-TO-END ARCHITECTURE

After analysing the different experiments, the final end-to-end DL architecture named as *hitMusicNet* for predicting the popularity of a song is composed by two main stages: a deep autoencoder and a deep neural network whose parameters correspond to *Model 1-A* and are depicted in Table 3.

VII. CONCLUSION

A complete overview of an end-to-end DL architecture for predicting the popularity of a certain song has been described. In addition, this paper has emphasized the use of multimodal information including both text and audio descriptors to enhance the feature vector basis that represents the song.

Moreover, a dataset named as *SpotGenTrack* Popularity Dataset has been constructed based on both *Spotify* and *Genius* platforms to create a multimodal database composed by meta-data, high-level and low-level audio features as well as text descriptors. Additionally, the preview URLs and lyrics are also included to enable the extraction of new features that may lead to a higher precision when predicting popularity in future experiments.

From an audio processing perspective, a set of features such as the *Tonnetz* or the Chromagram representations were considered to better represent the audio together with the Mel-spectrogram coefficients and the energy of the audio signal. On the other hand, regarding text analysis, a set of significant text features including a sentence similarity and a wealth-vocabulary metrics were incorporated into the analysis and allowed us to obtain a more consistent feature vector to feed the proposed architecture.

Furthermore, the main elements involved in the creation of *HitMusicNet* have been described including a deep autoencoder network denoted as *MusicAENet* as well as a deep neural network named *MusicPopNet*. The former is used to improve the quality of the feature vector by compressing the information whereas the latter is in charge of performing either the classification or the regression task.

Finally, several experiments from both classification and regression perspectives were conducted using different configurations of the aforementioned components to obtain the most adequate solution to address the problem. In addition, it has been demonstrated that the results are considerably better than in previous studies.

Regarding future work, many improvements will be adopted including the inclusion of automatic feature extraction procedures via Convolutional Neural Networks (CNN) and Word Embedding representations for audio and text respectively to improve the end-to-end architecture. The main objective consists in having independent branches for each modality and then concatenate them all before being introduced into the proposed *MusicAENet* component. Finally, the code to test the models and to compare the architecture with new innovative solutions is available at this **Github** repository.

REFERENCES

[1] P. Achananuparp, X. Hu, and X. Shen, "The evaluation of sentence similarity measures," in *Proc. Int. Conf. Data Warehousing Knowl. Discovery*. Berlin, Germany: Springer, 2008, pp. 305–316.

[2] F. Argenti, P. Nesi, and G. Pantaleo, "Automatic transcription of polyphonic music based on the Constant-Q bispectral analysis," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 19, no. 6, pp. 1610–1630, Aug. 2011.

[3] M. A. Azis, A. Hamid, A. Fauzi, E. Yulianto, V. Riyanto, Ridwansyah, and Sfenrianto, "Information retrieval system in text-based skripsi document search file using vector space model method," *J. Phys., Conf. Ser.*, vol. 1367, no. 1, 2019, Art. no. 012016.

[4] T. Bergstrom, K. Karahalios, and J. C. Hart, "Isochords: Visualizing structure in music," in *Proc. Graph. Interface*, 2007, pp. 297–304.

[5] T. Bertin-Mahieux, D. P. Ellis, B. Whitman, and P. Lamere, "The million song dataset," in *Proc. ISMIR*, vol. 2, 2011, p. 10.

[6] K. Bischoff, C. S. Firan, M. Georgescu, W. Nejdl, and R. Paiu, "Social knowledge-driven music hit prediction," in *Proc. Int. Conf. Adv. Data Mining Appl.* Berlin, Germany: Springer, 2009, pp. 43–54.

[7] D. Bogdanov, J. Serra, N. Wack, P. Herrera, and X. Serra, "Unifying low-level and high-level music similarity measures," *IEEE Trans. Multimedia*, vol. 13, no. 4, pp. 687–701, Aug. 2011.

[8] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proc. COMPSTAT*. Springer, 2010.

[9] R. Caruana, S. Lawrence, and C. L. Giles, "Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping," in *Proc. Adv. Neural Inf. Process. Syst.*, 2001, pp. 402–408.

[10] R. Cohn, "Neo-riemannian operations, parsimonious trichords, and their 'tonnetz' representations," *J. Music Theory*, vol. 41, no. 1, pp. 1–66, 1997.

[11] R. Dhanaraj and B. Logan, "Automatic prediction of hit songs," in *Proc. ISMIR*, 2005, pp. 488–491.

[12] T. Dozat, "Incorporating nesterov momentum into adam," Stanford Univ., Stanford, CA, USA, Tech. Rep., 2016.

[13] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. Hoboken, NJ, USA: Wiley, 2012.

[14] D. P. Ellis, B. Whitman, T. Jehan, and P. Lamere, "The echo nest musical fingerprint," Columbia Univ., Columbia, SC, USA, Tech. Rep., 2010.

[15] J. Fan and M. Casey, "Study of Chinese and UK hit songs prediction," in *Proc. Int. Symp. Comput. Music Multidisciplinary Res.*, 2013, pp. 640–652.

[16] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognit. Lett.*, vol. 27, no. 8, pp. 861–874, Jun. 2006.

[17] J. T. Foote, "Content-based retrieval of music and audio," *Proc. SPIE*, vol. 3229, pp. 138–147, Oct. 1997.

[18] R. Gasser, L. Rossetto, and H. Schuldt, "Multimodal multimedia retrieval with Vitrivr," in *Proc. Int. Conf. Multimedia Retr. (ICMR)*, 2019, pp. 391–394.

[19] R. Gasser, L. Rossetto, and H. Schuldt, "Towards an all-purpose content-based multimedia information retrieval system," 2019, *arXiv:1902.03878*. [Online]. Available: http://arxiv.org/abs/1902.03878

[20] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. 13th Int. Conf. Artif. Intell. Statist.*, 2010, pp. 249–256.

[21] G. Gwardys and D. Grzywczak, "Deep image features in music information retrieval," *Int. J. Electron. Telecommun.*, vol. 60, no. 4, pp. 321–326, Dec. 2014.

[22] P. Hamel and D. Eck, "Learning features from music audio with deep belief networks," in *Proc. ISMIR*, Utrecht, The Netherlands, vol. 10, 2010, pp. 339–344.

[23] J. Han and C. Moraga, "The influence of the sigmoid function parameters on the speed of backpropagation learning," in *Proc. Int. Workshop Artif. Neural Netw.* Berlin, Germany: Springer, 1995.

[24] K. Han, Y. Wang, C. Zhang, C. Li, and C. Xu, "Autoencoder inspired unsupervised feature selection," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2018, pp. 2941–2945.

[25] C. Harte, M. Sandler, and M. Gasser, "Detecting harmonic change in musical audio," in *Proc. 1st ACM Workshop Audio Music Comput. Multimedia (AMCMM)*, 2006, pp. 21–26.

[26] 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.

[27] D. Herremans, D. Martens, and K. Sörensen, "Dance hit song prediction," *J. New Music Res.*, vol. 43, no. 3, pp. 291–302, Sep. 2014.

[28] S. Hochreiter, Y. Bengio, P. Frasconi, and J. Schmidhuber, "Gradient flow in recurrent nets: The difficulty of learning long-term dependencies," Technische Univ. München, Univ. Montréal, IDSIA, Univ. Florence, Florence, Italy, Tech. Rep., 2001.

[29] D.-N. Jiang, L. Lu, H.-J. Zhang, J.-H. Tao, and L.-H. Cai, "Music type classification by spectral contrast feature," in *Proc. IEEE Int. Conf. Multimedia Expo*, vol. 1, Aug. 2002, pp. 113–116.

[30] L. Jing, B. Liu, J. Choi, A. Janin, J. Bernd, M. W. Mahoney, and G. Friedland, "DCAR: A discriminative and compact audio representation for audio processing," *IEEE Trans. Multimedia*, vol. 19, no. 12, pp. 2637–2650, Dec. 2017.

[31] S. Karsoliya, "Approximating number of hidden layer neurons in multiple hidden layer BPNN architecture," *Int. J. Eng. Trends Technol.*, vol. 3, no. 6, pp. 714–717, 2012.

[32] I. Karydis, A. Gkiokas, and V. Katsouros, "Musical track popularity mining dataset," in *Proc. Int. Conf. Artif. Intell. Appl. Innov. (IFIP)*. Cham, Switzerland: Springer, 2016, pp. 562–572.

[33] I. Karydis, A. Gkiokas, V. Katsouros, and L. Iliadis, "Musical track popularity mining dataset: Extension & experimentation," *Neurocomputing*, vol. 280, pp. 76–85, Mar. 2018.

[34] T. Kenter and M. de Rijke, "Short text similarity with word embeddings," in *Proc. 24th ACM Int. Conf. Inf. Knowl. Manage. (CIKM)*, 2015, pp. 1411–1420.

[35] B. Kim and B. Pardo, "Improving content-based audio retrieval by vocal imitation feedback," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 4100–4104.

[36] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980

[37] J. Lee and J.-S. Lee, "Predicting music popularity patterns based on musical complexity and early stage popularity," in *Proc. 3rd Ed. Workshop Speech, Lang. Audio Multimedia (SLAM)*, 2015, pp. 3–6.

[38] J. Lee and J.-S. Lee, "Music popularity: Metrics, characteristics, and audio-based prediction," *IEEE Trans. Multimedia*, vol. 20, no. 11, pp. 3173–3182, Nov. 2018.

[39] Y. Li and Y. Yuan, "Convergence analysis of two-layer neural networks with ReLU activation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 597–607.

[40] B. Logan, "MEL frequency cepstral coefficients for music modeling," in *Proc. ISMIR*, vol. 270, 2000, pp. 1–11.

[41] L. Lu, H. Jiang, and H. Zhang, "A robust audio classification and segmentation method," in *Proc. 9th ACM Int. Conf. Multimedia (MULTIMEDIA)*, 2001, pp. 203–211.

[42] W. Ma and T. Suel, "Structural sentence similarity estimation for short texts," in *Proc. FLAIRS Conf.*, 2016, pp. 232–237.

[43] L. V. D. Maaten and G. Hinton, "Visualizing data using T-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.

[44] A. Martins and R. Astudillo, "From softmax to sparsemax: A sparse model of attention and multi-label classification," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 1614–1623.

[45] A. C. North, A. E. Krause, L. P. Sheridan, and D. Ritchie, "Energy, popularity, and the circumplex: A computerized analysis of emotion in 143,353 musical pieces," *Empirical Stud. Arts*, vol. 36, no. 2, pp. 127–161, May 2017.

[46] J. C. Nunes, A. Ordanini, and F. Valsesia, "The power of repetition: Repetitive lyrics in a song increase processing fluency and drive market success," *J. Consum. Psychol.*, vol. 25, no. 2, pp. 187–199, Apr. 2015.

[47] F. Pachet and P. Roy, "Hit song science is not yet a science," in *Proc. ISMIR*, 2008, pp. 355–360.

[48] F. Pachet and C. Sony, "Hit song science," in *Music Data Mining*. Boca Raton, FL, USA: Chapman & Hall, 2012.

[49] J. Pham, E. Kyauk, and E. Park, "Predicting song popularity," Dept. Comput. Sci., Stanford Univ., Stanford, CA, USA, Tech. Rep., 2016, vol. 26.

[50] J. Pons and X. Serra, "Randomly weighted CNNs for (music) audio classification," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 336–340.

[51] A. Qayyum, S. M. Anwar, M. Awais, and M. Majid, "Medical image retrieval using deep convolutional neural network," *Neurocomputing*, vol. 266, pp. 8–20, Nov. 2017.

[52] I. Rivals and L. Personnaz, "Neural-network construction and selection in nonlinear modeling," *IEEE Trans. Neural Netw.*, vol. 14, no. 4, pp. 804–819, Jul. 2003.

[53] S. Robertson, "Understanding inverse document frequency: On theoretical arguments for IDF," *J. Document.*, vol. 60, no. 5, pp. 503–520, Oct. 2004.

[54] M. Sahidullah and G. Saha, "Design, analysis and experimental evaluation of block based transformation in MFCC computation for speaker recognition," *Speech Commun.*, vol. 54, no. 4, pp. 543–565, May 2012.

[55] R. N. Shepard, "Circularity in judgments of relative pitch," *J. Acoust. Soc. Amer.*, vol. 36, no. 12, pp. 2346–2353, Dec. 1964.

[56] S. Sigtia and S. Dixon, "Improved music feature learning with deep neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2014, pp. 6959–6963.

[57] D. F. Silva, C.-C.-M. Yeh, Y. Zhu, G. E. A. P. A. Batista, and E. Keogh, "Fast similarity matrix profile for music analysis and exploration," *IEEE Trans. Multimedia*, vol. 21, no. 1, pp. 29–38, Jan. 2019.

[58] A. Singhi and D. G. Brown, "Hit song detection using lyric features alone," in *Proc. Int. Soc. (MIR)*, 2014, pp. 3–8.

[59] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.

[60] V. Taktakishvili, A. Ovchinnikov, O. Popov, and V. Abramov, "Analysis and processing of audio signals using complex form representation," in *Proc. 24th Conf. Open Innov. Assoc. FRUCT*, 2019, p. 105.

[61] 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.

[62] S. Wang, Z. Ding, and Y. Fu, "Feature selection guided auto-encoder," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 2725–2731.

[63] M. Xu, L.-Y. Duan, J. Cai, L.-T. Chia, C. Xu, and Q. Tian, "HMM-based audio keyword generation," in *Proc. Pacific-Rim Conf. Multimedia*. Berlin, Germany: Springer, 2004, pp. 566–574.

[64] Y. Yu, S. Tang, F. Raposo, and L. Chen, "Deep cross-modal correlation learning for audio and lyrics in music retrieval," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 15, no. 1, pp. 1–16, Feb. 2019.

[65] M. D. Zeiler, "ADADELTA: An adaptive learning rate method," 2012, *arXiv:1212.5701*. [Online]. Available: http://arxiv.org/abs/1212.5701

[66] X. Zeng and T. R. Martinez, "Distribution-balanced stratified cross-validation for accuracy estimation," *J. Experim. Theor. Artif. Intell.*, vol. 12, no. 1, pp. 1–12, Jan. 2000.

[67] Z. Zhang and M. Sabuncu, "Generalized cross entropy loss for training deep neural networks with noisy labels," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 8778–8788.

DAVID MARTÍN-GUTIÉRREZ (Fellow, IEEE) received the bachelor's degree in audio-visual system engineering from Carlos III University, and the master's degree in signal processing and machine learning from the Universidad Politécnica de Madrid (UPM). He worked at the Banking Department, Everis S.L., as a Software Developer, implementing trading applications. Later on, he worked at Ixion Industry and Aerospace as a Data Sensor Fusion Engineer, where he implemented fusion algorithms for UAV's. He is currently with the Visual Telecommunications Applications Group (GATV) as a Predoctoral Researcher, developing artificial intelligent algorithms in several national and EU projects. His final project consisted of the study and the implementation of Bayesian Inference Algorithms in state-space systems.

GUSTAVO HERNÁNDEZ PEÑALOZA (Member, IEEE) received the degree in telecom engineering from the Universidad Santo Tomás, in 2007, and the M.Sc. degree in telecommunication technologies, system and networks from the Universidad Politécnica de Valencia (UPV), in 2009. He is currently pursuing the Ph.D. degree with the Visual Telecommunications Applications Group (GATV), Universidad Politécnica de Madrid (UPM). From 2010 to 2013, he worked as an Associate Research Fellow at the "Universidad de Valencia" (UV). He currently works for the research group in GATV, UPM. He has participated in different technical developments in several national and EU projects.

ALBERTO BELMONTE-HERNÁNDEZ (Member, IEEE) received the degree in telecommunication engineering and the master's degree, focused on communication systems, from the Universidad Politécnica de Madrid" (UPM), in 2014 and 2016, respectively, where he is currently pursuing the Ph.D. degree with the Visual Telecommunications Applications Group (GATV). He has worked at Everis Spain SL using Liferay Programming Tool and JAVA programming language to deploy web services and web pages. He is currently working for the research group in the Visual Telecommunications Applications Group (GATV), Universidad Politécnica de Madrid (UPM). His main interests are the new communications technologies, the Internet of Things (IoT), sensors, cameras, and wireless communications. He is actively working on machine/deep learning algorithms applied to sensors and image. He has been developing technical parts in national and EU projects.

FEDERICO ÁLVAREZ GARCÍA (Member, IEEE) received the degree (Hons.) in telecom engineering and the Ph.D. degree *(cum laude)* from the Universidad Politécnica de Madrid (UPM), in 2003 and 2009, respectively. Since 2003, he has been working for the research group in the Visual Telecommunications Applications Group (GATV), UPM, where he is currently working as an Assistant Professor. He has been participating with different managerial and technical responsibilities in several national and EU projects, being a coordinator of five EU projects in the last six years. He had participated in national and international standardization fora (DVB, CENELEC TC206, etc.) He is a member of the program committee of several scientific conferences and the author and coauthor of over 60 papers and several books, book chapters, and holds patents in the field of ICT networks and audiovisual technologies.