PRESTO: A Recommender of Musical Collaborations Based on Heterogeneous Graph Neural Networks

Fernando Terroso-Saenz[1]*, Jesús Soto[2], Andrés Muñoz[3], Philippe Roose[4]

[1] Technical University of Cartagena, Cartagena (Spain)
[2] Universidad Católica de Murcia, Murcia (Spain)
[3] University of Cádiz, Cadiz (Spain)
[4] Université de Pau et des Pays de l'Adour, Anglet (France)

* Corresponding author: fernando.terroso@upct.es

Received 15 March 2024 | Accepted 24 January 2025 | Early Access 12 March 2025

Abstract

The music industry is now more complex and competitive than ever before. In recent years, the search for collaborations with other artists has become a common strategy for musicians to maintain their presence in the sector. Besides, existing music streaming services such as Spotify have exposed large data feeds that can be used to develop innovative services within the realm of music. In this context, the present work introduces PRESTO, a novel recommendation system to suggest musicians for new collaborations with other artists by means of an ensemble of Graph Neural Networks. The system is fed with a heterogeneous graph representing the time evolution and the stationary aspects of a musician's career. Finally, the proposal has been evaluated with a dataset comprising more than 200,000 artists, with an average F1 score above 0.75.

Keywords

Artificial Intelligence Tools, Graph Neural Network, Heterogeneous Graph, Musical Collaborations, Recommender System.

DOI: 10.9781/ijimai.2025.03.004

I. Introduction

In the music scene, collaboration among artists is considered one of the most impressive driving factors that guide the production of new songs. These artistic collaborations allow creating songs spanning multiple genres and styles, and therefore reaching a wider audience. In that sense, there are studies that already highlight the unexpected and complementary ideas that emerge during a musical collaboration [1]. Hence, the search for collaborations with other artists have become a common strategy in most musicians' careers to maintain their presence in the complex and competitive music market [2].

As a matter of fact, the work in [3] discusses how BTS (Korean Boy Band) benefits from collaborations with Western artists. Other studies present a social network used by rappers and analyze how rap music is structured according to rapper collaborations [4]. Moreover, some recent studies report that some artists saw an increase of at least 10% in their Spotify streams when they participated in crossover collaborations from 2012 to 2023 [5]. From the fans' perspective, some experts emphasize that collaborations, whether big or small, help artists grow beyond their fanbase and cross borders [6]. Indeed, some studies have noted that musical collaborations already accounted for over 40% of the songs in the Billboard Hot 100 songs by 2020 [7]. Additionally, other works state that the number of collaborations in that same ranking increased by a factor of 5 from 1988 to 2018 [8].

At the same time, the digital era has brought the opportunity to listen to any kind of music from a vast number of musicians through many different streaming services such as Spotify, YouTube, or Apple Music. In fact, Spotify includes more than 11 million artists in its catalogue[1] whereas Apple Music claims that its catalogue comprises around 5 million recording artists[2].

These streaming services represent a wealth of interesting data sources, motivating an emerging line of research within the data science community. Some of the analyses in this line have mainly followed a user-centric view, focusing on solutions for end-users, such as music recommendation [9], [10] or sentiment analysis [11]. Regarding the analysis of musical collaborations, these artist-centric proposals usually follow two prominent lines of research, namely (1) exploratory analyses of the main factors that have influenced previous collaborations among musicians [12], [13] and (2) approaches for hit-song prediction that take into account some collaboration features among the involved artists [14].

Nevertheless, few works have exploited such data to help musicians discover other artists whose genres or styles may fit with their own and lead to new songs. One possible reason for this scarcity of works might be the fact that the *corpus* of end users is much larger

[1] https://www.demandsage.com/spotify-stats/

[2] https://artists.apple.com/support/1124-apple-music-insights-royalty-rate

Please cite this article as:
F. Terroso-Saenz, J. Soto2, A. Muñoz3, P. Roose. PRESTO: A Recommender of Musical Collaborations Based on Heterogeneous Graph Neural Networks, International Journal of Interactive Multimedia and Artificial Intelligence, (2025), http://dx.doi.org/10.9781/ijimai.2025.03.004

than that of musicians, making this market much more appealing for the development of applications. In fact, the aforementioned music streaming services give access to a large catalogue of artist information that could be used to perform such a *music compatibility* task in the form of innovative recommendation systems for artists and music companies.

In this context, the present work introduces *PRESTO*, a graPh neuRal nEtwork for muSical collaboraTions recOmmendation. This graph-based approach captures the relationships among the involved actors in a more natural way than other deep learning solutions such as recurrent neural networks or multi-layer perceptrons. As Fig. 1 shows, the system is fed with existing songs created by artists either individually or collaboratively, along with other features related to these songs and the artists themselves, such as the danceability of the song or the artist's popularity. Based on this information, PRESTO suggests new collaborations among pairs of artists who have not worked together before. As such, PRESTO has been designed to help musicians find suitable partners to produce new tracks.

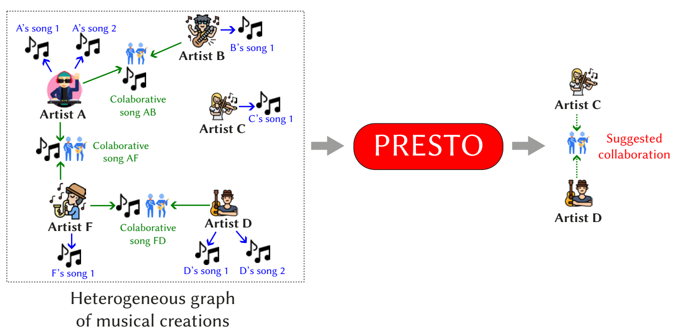


Fig. 1. Overview of PRESTO. The system takes as input the existing collaborations and songs of musicians and recommends unseen collaborations among them (like the one between *Artist C* and *Artist D* in the image).

The design of this system relies on the hypothesis that the career of a musician usually follows some type of evolution regarding their songs. For example, a well-known Hollywood composer, Hans Zimmer, emerged in 2017 as a headliner of the Coachella Festival, one of the most popular rock and pop music festivals[3]. Hence, an artist feature-based recommendation must consider not only *which* previous collaborations the artist has been involved in but also *when* they occurred, that is, the artist's evolution in musical terms. This calls for a hybrid deep learning architecture combining different types of convolutional and recurrent layers to capture the *static* and *time-evolving* features of artists.

To do so, the proposed system is based on a heterogeneous graph to model the relationships among artists, songs, music genres, their latent features and their temporal evolution. Next, an ensemble of different types of Graph Neural Networks (GNNs) are used to anticipate future relationships by means of a link-prediction approach. To feed the model, we have used a large dataset from Spotify and Last-FM that includes collaborations among multiple artists covering a 17-year period. Then, the predicted new links can be used as recommended pairwise collaborations among artists to better define the future development of an artist's career.

The remainder of the paper is structured as follows. Section II gives an overview of existing trends and techniques for analyzing music streaming data. Then, Section III introduces the data extraction and curation process from the target feeds. Section IV describes the inner architecture of PRESTO. Then, Section V presents the evaluation of the proposed recommender. Lastly, Section VI summarizes the main conclusions and potential future research lines motivated by this work.

II. Related Work

The analysis of streaming music data has been the subject of different lines of research within the field of music-related recommendation systems [15]. First, there are works that have focused on analyzing the built-in song recommenders included in platforms such as Spotify or Apple Music from different points of view. For instance, authors in [16] evaluate the satisfaction level of the Spotify recommender among different users, whereas the work in [17] outlines the gender bias of some of these recommendation systems.

Another prominent course of action has been the development of ad-hoc recommendation systems to suggest new songs based on listeners' preferences or contextual factors. In that sense, the usage of deep learning techniques is remarkable due to their ability to extract latent factors from music items in an isolated or sequential manner [18]. Hence, graph neural networks (GNNs) have been proposed to analyze hypergraphs capable of encoding complex relationships among listeners and songs [19]. A similar GNN approach with an attention mechanism was proposed in [20].

Regarding other types of neural networks, a bidirectional gated Recurrent Neural Network (RNN) was applied in [21] to detect the current physical activity of a user and then suggest a new music file whose *tempo* better fits the intensity of the activity. A Multilayer Perceptron (MLP) is the core architecture of T-RECSYS [22], a hybrid song recommendation tool that combines content-based and collaborative filtering.

Concerning Convolutional Neural Networks (CNNs), they have been applied in [23] to capture the listener's mood based on their facial expression, suggesting new songs accordingly. CNNs have also been applied in the context-aware system described in [24] for background music recommendation in a smart home. Beyond deep-learning, other models such as Hidden Markov models have also been proposed for music recommendation [25].

All the aforementioned works focus on providing recommendations to end users. From an artist's perspective, an important line of research in the music sector has been the *Hit Song Science* [26], whose goal is to predict the *popularity* of a song before it hits the market. Some works in this field rely on Machine Learning models for this task. Thus, the work in [27] analyzes songs' audio features (such as rhythm and instrumentation) and metadata to discover past successful music trends and then replicate them for future songs. The authors in [28] add a sentiment analysis of the lyrics to the study of audio features for a more accurate prediction. Both works utilize and compare logistic regression, Naive Bayes and Random Forest models in the analysis, obtaining an accuracy of around 50% for all models. Neural networks have also been applied in this context. For instance, the authors of [29] define an MLP that combines low- and high-level features of a song along with its release date to provide an estimate of its peak position in the charts. Similarly, MLPs have been used for pairwise hit-song prediction to forecast which song will be more popular between a given pair [30].

Concerning recommender systems for suggesting collaborations among artists, the authors in [12] perform an analysis on a graph modeling the collaborations among 2,152 artists. Based on different centrality and similarity metrics, they extracted three different collaboration clusters representing diverse, regular, and absent collaborations. Another study focused on identifying the main factors that guide the composition of collaborative songs [13].

Our approach also intends to provide a novel recommendation service for music artists. However, unlike previous work, PRESTO

[3] https://www.theatlantic.com/entertainment/archive/2017/02/howhans-zimmer-became-a-rockstar/516912/

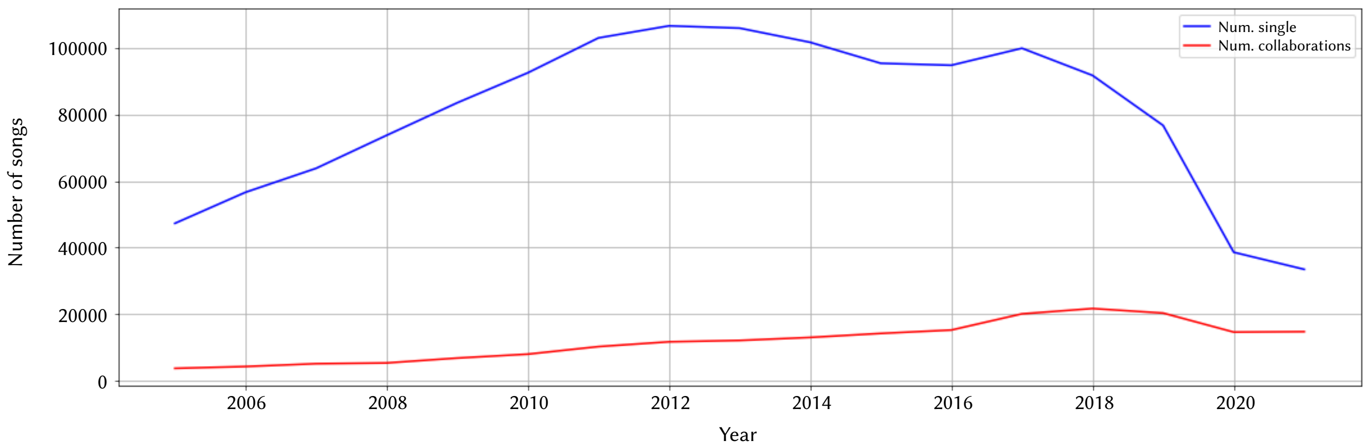


Fig. 2. Evolution in the number of tracks per year in the dataset based on their number of artists. The red line shows the collaborative tracks, whereas the blue line shows the tracks featuring a single artist.

focuses on recommending collaborations among musicians to work together on new songs. That is, our approach could be regarded as a first step before hit-song predictors are applied. Hence, it also goes beyond the exploratory works for musical collaborations described above. To the best of our knowledge, this is the first attempt to develop this type of recommender following a deep-learning approach.

III. Composition of the Heterogeneous Music Graph for PRESTO

This section describes the datasets used for developing PRESTO along with an explanatory analysis of them. Next, the creation of a heterogeneous music graph used to train PRESTO is explained.

A. Datasets

The development of PRESTO relies on two different raw music feeds. On the one hand, we have used the *Top 200* daily rankings released by Spotify, which contain the 200 most played songs across 69 different countries per day[4]. By means of an ad-hoc crawler, daily rankings were extracted for each country for a two-year period from July 17, 2020, to October 11, 2022. As a result, 14,816 unique songs were extracted from 2,569 artists. On the other hand, we have also processed the Last-FM (LFM) 2b dataset [31]. This dataset comprises 2,378,113 Spotify tracks from 266,479 artists covering a 15-year period (February 18, 2005 – March 20, 2020) in Last.fm, a well-known online music service. It is important to note that these tracks differ from those in the Top-200 ranking feed mentioned earlier.

Next, we fused both feeds by removing duplicates, resulting in a single dataset D comprising 1,354,932 tracks, 259,698 unique artists and 2,166 music genres covering a 17-year period from February 18, 2005, to October 11, 2022. The artists included in this dataset will be represented as the set $\mathcal{A}$, the genres as $\mathcal{G}$ and the songs as $\mathcal{S}$ in the following sections.

Finally, for each song in $\mathcal{D}$, we extracted 10 features (represented as $\mathcal{F}_s$ from now on) from the Spotify Developer Platform (SDP)[5]. These features are related to the song's audio properties (loudness, speechiness and instrumentalness), context (liveness and acousticness) and mood (danceability, valence, energy, and tempo), along with its release date as the tenth feature[6]. Moreover, for each individual artist, we extracted 3 features (represented as $\mathcal{F}_a$), namely 1) their number of followers on Spotify, 2) associated music genres, and 3) popularity score. This last score ranges from 0 to 100 and is computed mathematically by Spotify.

B. Exploratory Study of the Data

In order to define a heterogeneous graph that properly captures the relationships between the different elements involved in the recommendation process, namely artists, songs, and genres, we performed an exploratory study on the aforementioned dataset whose main findings are stated next. These three elements not only allow us to capture the artists involved in a collaboration but also the features of these collaborations.

1. Evolution in the Number of Collaborations

To begin with, we split $\mathcal{D}$ into two disjoint groups, namely songs with a single artist and songs with multiple singers. Fig. 2 shows the evolution of the number of tracks of these two groups during the whole period of study. As observed, the number of collaborative tracks steadily increases throughout the whole period of study. However, this steady increment is not observed for tracks with a single artist, where a slight decline has been noted since 2012. This is consistent with the findings of previous works that reported an increase in this type of collaborations in the worldwide music market [32].

2. Distribution of Music Genres

We also studied the discrepancy between the music genres of an artist and those of the musicians they have collaborated with. Fig. 3a shows the connections among genres by considering the artists individually. As observed, many artist labeled as *pop* are frequently also labeled with the *indie*, *punk* or *rock* genres as well. Moreover, the *alternative* genre is usually collocated with the *rock* genre.

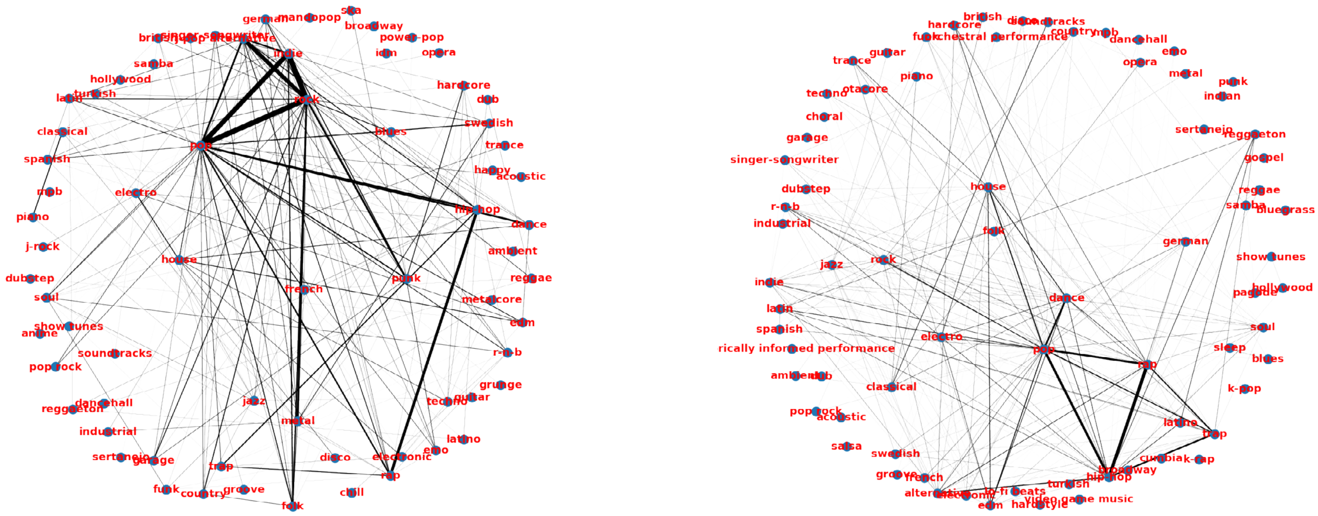
Fig. 3b shows a similar graph but showing the connections between genres based on the collaborations among artists. From this graph, it can be seen that *pop* artists frequently collaborate with other *rap* or *hip-hop* artists. Similarly, another strong connection occurs between *rap* and *hip-hop* artists. This makes sense due to the *conceptual* similarities between these genres.

Lastly, Fig. 4 shows the Shannon Entropy [33] of the distribution of genres in the labeling of each artist individually and in the collaboration among musicians. It can be observed that the highest entropy occurs in less common, rare, or local genres such as *honky-tonk*, *iranian* or *party*. However, the entropy of the most well-known genres is very low. For example, the entropy of the *pop*, *rock* and *hip-hop* genres was 0.1219, 0.1304 and 0.0301, respectively. This low entropy in most of

[4] https://charts.spotify.com/charts/overview/global

[5] https://developer.spotify.com/discover/

[6] We assumed that the release date of a song corresponded to that of the album on which it appeared.



(a) Distribution of genres per artist.

(b) Distribution of genres per collaboration.

Fig. 3. Analysis of the distribution of genres per artist and collaboration. The nodes represent the subset of the most frequent music genres, and the width of each edge is proportional to the frequency of each pair of genres.

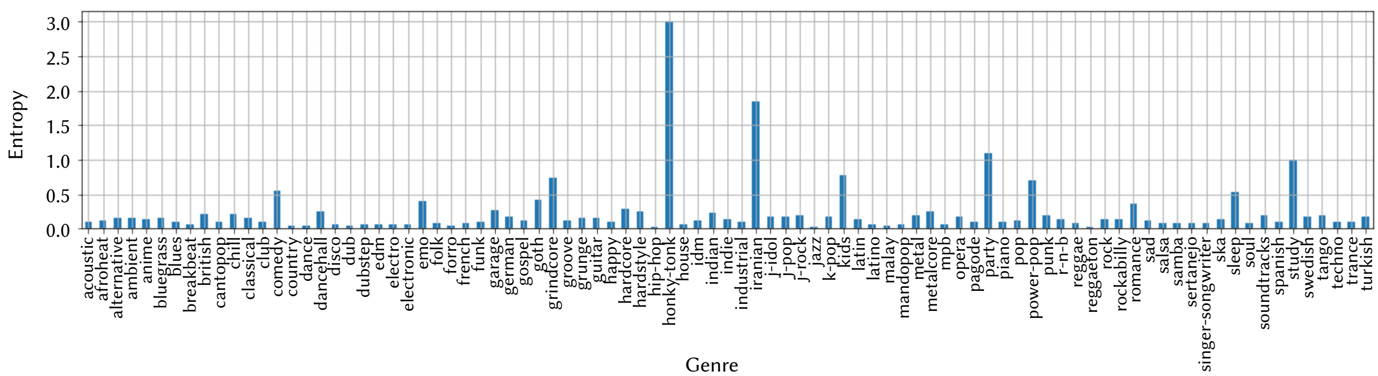


Fig. 4. Entropy per music genre based on their distribution per artist and musical collaborations.

the common genres reveals that artists with similar genres tend to collaborate more frequently. Therefore, the resulting heterogeneous graph should incorporate this genre feature in some manner.

3. Distribution of Songs' Audio Features

Another analysis explored the differences in terms of audio features between the songs an artist creates solo and those performed with other musicians. Fig. 5 shows the average variation in the 9 audio features between songs created by a single artist and songs in which the same artist has collaborated with other musicians.

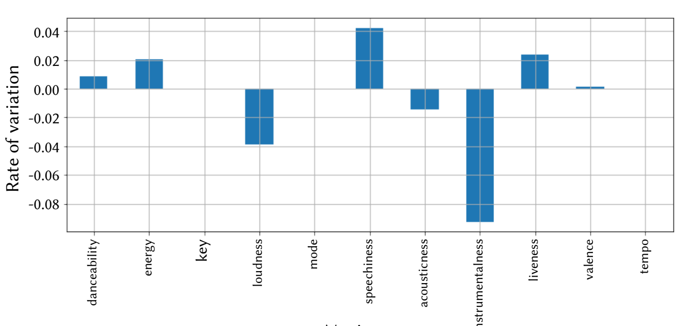


Fig. 5. Average variation of the audio features of an artist's songs when they are created in a collaborative manner with respect to the tracks that are created by the artist solo.

As observed, the variations tend to be very small in almost all features. Notably, the highest shift occurs in the instrumentalness feature, where collaborative songs tend to have a value only 0.08 lower than solo songs. This finding reveals that the audio features of the songs created by an artist solo must be taken into account when recommending collaborations due to their similarities with the ones they could create with other musicians. Indeed, these features encode latent features that could be used to measure the similarity among artists in order to assess a potential collaboration.

4. Evolution of the Songs' Audio Features

One important assumption of this work is that the recommendation of potential collaborations is not solely defined by previous collaborations or the songs that an artist has created, but is also affected by the temporal evolution of such compositions. In order to evaluate the evolution of a musician's career in terms of audio features, we compose a timeseries t_a^f for each artist a and song feature $f, f \in \mathcal{F}_s$. In this manner, each sequence describes the evolution of the artists' audio features over time, based on their songs. Note that this sequence could be obtained by sorting the songs based on their release date. Then, we applied the Augmented Dickey-Fuller Test [34] on each sequence. The null hypothesis of this test states that the time series contains a unit root and thus it is non-stationary. The alternative hypothesis is that the time series does not contain a unit root, indicating that it is stationary. Fig. 6 shows the distribution of stationary and non-stationary timeseries per feature according to this test.

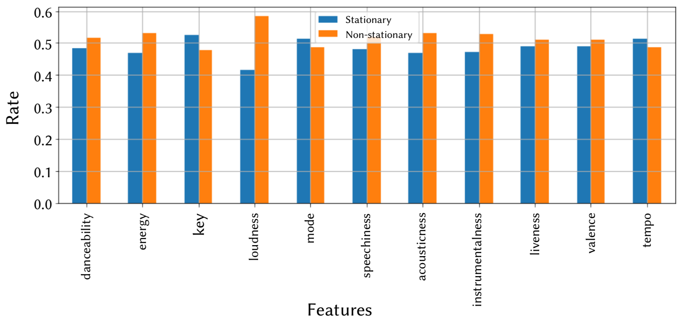


Fig. 6. Distribution of the features timeseries t_a^f based on their seasonality according to the Dickey-Fuller test with a p-value of 0.01.

As observed in this figure, the rate of non-stationary timeseries was rather high in almost all features, with a rate close to 50%. This reflects that, in many cases, musicians tend to evolve over time as the audio features of their songs do not remain static. Hence, the recommender system should consider this variation in evolution patterns as a relevant latent feature for establishing connections between musicians.

5. Impact of the Artists' Popularity

Regarding the artists' collected features, we computed the Pearson correlation between the popularity of an artist and their number of followers. This was done to determine whether a positive or negative relationship exists between these features. The obtained value was rather low, 0.2066. Moreover, Fig. 7 shows the distribution of the artists' popularity across different collaborative songs in the dataset.

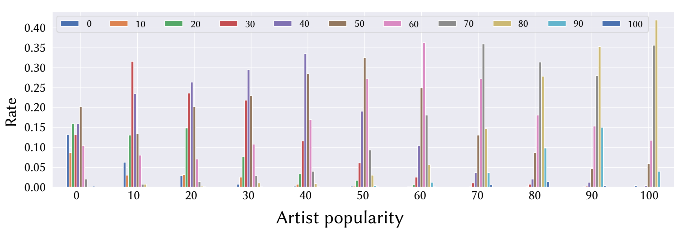


Fig. 7. Collocation of the artists popularity based on the observed collaborations. The y-axis shows the rate (up to 1.0) of artists with a popularity value reflected in the xaxis that work together with artists whose popularity level is indicated in the legend.

As observed, artists tend to collaborate with other artists who have a similar or slightly higher popularity score. For example, the pink bar at point 60 in the x-axis of Fig. 7 reflects that 35.4% of the artist with a popularity score of 60 collaborated with other artists who had the same score. Moreover, 40.3% of the artists with a popularity score of 100 collaborated with artists whose score was 80. This is consistent with the findings stated in the exploratory analysis of musical collaborations described in [12], which found that artists' popularity and follower scores were quite similar within each collaboration cluster identified in the study.

From this analysis, we concluded that the model should consider both popularity and number of followers as artist features due to their low correlation and the fact that popularity seems to be a key factor guiding potential collaborations with other musicians.

C. Heterogeneous Graph to Model Musical Collaborations

Bearing in mind all the findings stated in the previous sections, we eventually defined a heterogeneous graph to capture the relationships among the different elements involved in the study. The schema of this graph is depicted in Fig. 8. It comprises 4 different relationships among artists, their songs and their associated music genres. Note that the *has* and *is* relationships allow linking an artist and a song with their associated music genres, respectively. The *creates* relationship establishes the connection between one or several artists (in case of a collaboration) and the song they created as solo or in a collaboration. Finally, the *collaborates* edges define the relations between pairs of artists that have collaborated together on one or more songs.

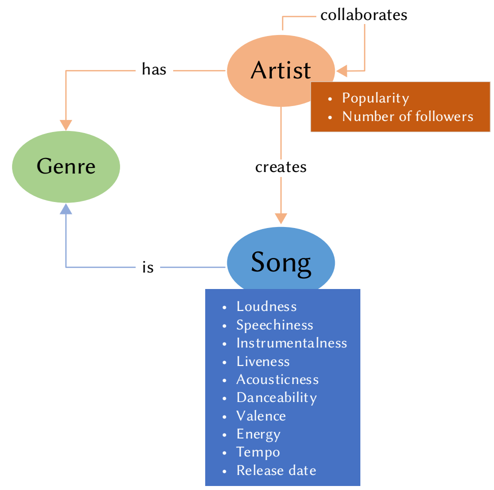


Fig. 8. Heterogeneous graph to encode the relationships and features between artists, songs and music genres. The colored rectangles contain the associated features of each song and artist node.

As an illustrative example, if artists A and B collaborate on a song S, then the graph will include a *collaborates* edge between A and B, and a *creates* edge from A to S and from B to S. As detailed in sec. IV, the latter type of relationship is the target edge that PRESTO will focus on predicting as a result of the recommendation process.

Given the dataset $\mathcal{D}$, this heterogeneous graph comprises 259,698 artists, 1.354,935 songs, and 2,166 genres, all represented as nodes. Moreover, it comprises 202,759 *has* relationships, 2,575,996 *is*, 1,509,557 *creates* and 337,910 *collaborates* ones.

IV. The PRESTO Application

This section describes in detail the architecture of the recommender system proposed in this work.

A. Temporal Split of the Heterogeneous Graph

In order to capture the musical evolution of all artists, we split the global dataset $\mathcal{D}$ (see sec A) into 30-day slices. The rationale behind this time granularity is that, as Fig. 9 shows, songs in the dataset $\mathcal{D}$ tend to be released at a relatively consistent rate from February to November. Note that the number of released songs was significantly higher in January compared to the other months, whereas it was lower in December. To smooth out such outliers, each slice $\mathcal{D}_i \subset \mathcal{D}, i = [1..12]$ comprised the songs with a release date falling within the time range from the 15th day of the i-th month to the 15th day of the following month. This approach ensured that songs released in January/ December were distributed across two different slices instead of one.

Next, we composed a heterogeneous time graph $\mathcal{G}_i^t$ (see sec. C) for each slice $\mathcal{D}_i$, comprising only nodes, edges, and features related to the songs included in the i-th time slice. This allowed us to compose an ordered sequence of time graphs $G_{seq}^t = \langle G_0^t \rightarrow G_1^t \rightarrow \ldots \mathcal{G}_i^t \rightarrow \mathcal{G}_{i+1}^t \rightarrow \ldots \mathcal{G}_{|200|}^t \rangle$ modeling the musical evolution of the artists during the 200 consecutive months of the study. For the sake of clarity, Fig. 10 shows the different types of edges in the time graph G_0^t. It is worth noticing the different densities of each sub-graph depending on the type of edges under consideration. This highlights the need for a model capable of handling such diversity in the prediction task.

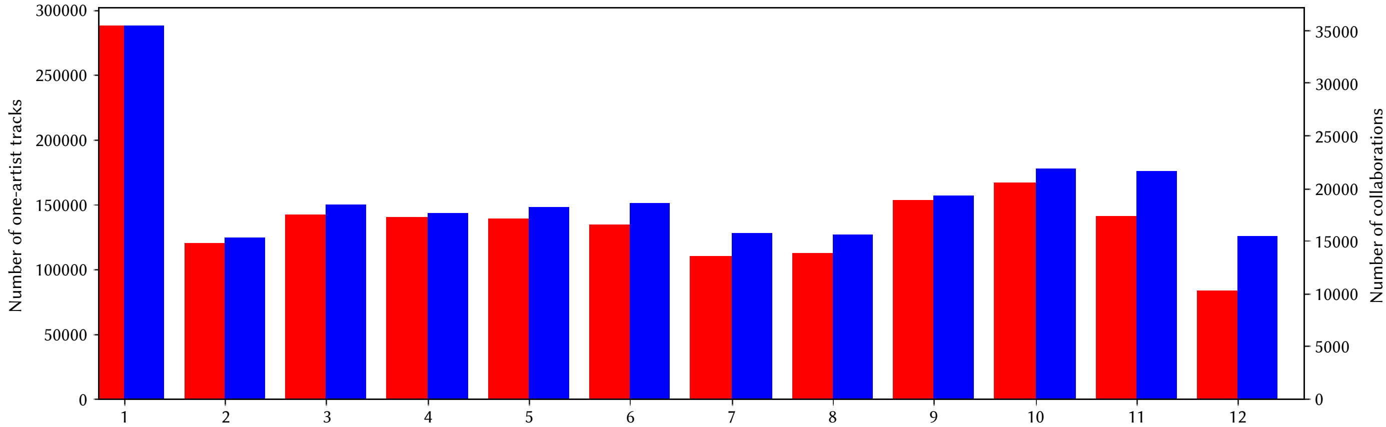
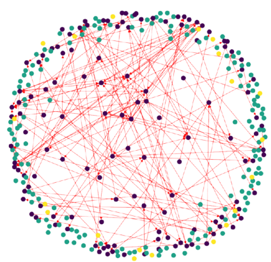
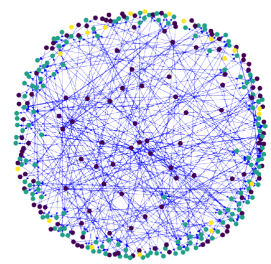


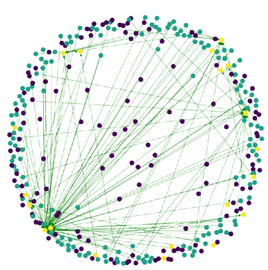
Fig. 9. Distribution of songs per month based on their release date. Collaborative songs are depicted in blue, while songs created by a single artist are shown in red.



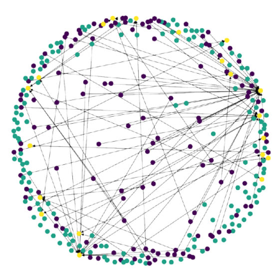
(a) *Collaborate* edges.



(b) *Produce* edges.



(c) *Has* edges.



(d) *Is* edges.

Fig. 10. Representation of the different edge types of the time graph $\mathcal{G}_0^t$ covering the first slice of study (2005/02/15-2005/03/15). The green circles represent the artists, the dark purple ones are songs, and the yellowones are music genres.

Additionally, we composed an aggregated graph, $\mathcal{G}_i^{agg}$, comprising the nodes, edges, and features related to all the songs from month 0 (the first month of study) up to the *i-th* month. Hence, a graph of this type represents the *static* snapshot of the target artists, as it does not distinguish when, for example, a particular artist created a song or collaborated with another musician.

The conjunction of the time $\mathcal{G}^t$ and aggregated $\mathcal{G}^{agg}$ graphs enables the provision of both the time-evolving and the static views of the artist ecosystem, which form the main hypothesis of this work.

B. Description of PRESTO's Inner Design

The inner layer structure of PRESTO is shown in Fig. 11. As observed, the architecture comprises 3 different layers, each one focusing on a different aspect in the processing pipeline and considering the heterogeneous nature of the input graphs.

The first layer takes as input the time and aggregated graph of a particular slice, $\mathcal{G}_i^t$ and $\mathcal{G}_i^{agg}$. Then, each graph is processed by a different graph operator.

On the one hand, $\mathcal{G}_i^t$ captures the time evolution of artists' production during the study period. For that reason, we used a heterogeneous Recurrent Neural Network (RNN) to handle this time-evolving sequence of graphs. More in detail, Gt i is processed by a version of the Integrated Graph Convolutional Long Short Term Memory for heterogeneous graphs (Hetero Conv-LSTM) model [35]. Basically, this version stacks a heterogeneous convolutional operator and an LSTM cell for each node type in the input graph, namely artist, song, and genre. Equations (1)-(5) refer to the computation of each gate for a node type η:

$$p_i^\eta = \sigma\left(W_{xp}^\eta \mathcal{G}_i^t + HC(h_{i-1}^t) + b_p^\eta\right) \tag{1}$$

$$f_i^\eta = \sigma\left(W_{xf}^\eta \mathcal{G}_i^t + HC(h_{i-1}^t) + b_f^\eta\right) \tag{2}$$

$$c_i^\eta = f_i^\eta \odot c_{i-1}^\eta + p_i^\eta \odot \tanh\left(W_{xc}\mathcal{G}_i^t + HC(h_{i-1}^t) + b_c^\eta\right) \tag{3}$$

$$o_{i,\eta}^\eta = \sigma\left(W_{xo}^\eta \mathcal{G}_i^t + HC(h_{i-1}^t) + b_o^\eta\right) \tag{4}$$

$$h_{i,\eta}^t = o^\eta \odot \tanh\left(c_i^\eta\right) \tag{5}$$

where p_i^η, f_i^η, c_i^η and o_i^η are the input, forget, state and output gates of the LSTM cell for a node type η at time step i, $W_{x\{p,f,c,o\}}^\eta$ and $W_{h\{p,f,c,o\}}^\eta$ are the weights of the fully connected layers of the node type η, η, $b_{\{p,f,c,o\}}^\eta$ are the bias terms of each layer, $\odot$ is Hadamard product and σ the sigmoid operator. Furthermore, HC refers to a heterogeneous graph convolution operator that uses a particular instance of the GraphSAGE model [36] for each edge type. Since the input graph $\mathcal{G}_i^t$ comprises 3 different node types, the final embedding $\mathcal{E}_i^t$ is just the concatenation of the $h_{i,artist}^t$, $h_{i,genre}^t$ and $h_{i,song}^t$. In this manner, it encodes the latent representation for each node type (artists, songs and genres) based on its temporal evolution. By adopting this heterogeneous approach, we allow the HeteroConvLSTM module to learn weights for each node and edge type. This is instrumental given the diverse sub-graphs comprising $\mathcal{G}_i^t$, as demonstrated in sec. A.

On the other hand, the aggregated graph $\mathcal{G}_i^{agg}$ is also processed by a heterogeneous version of the GraphSAGE model. In brief, the embedding generation in this model relies on an incremental aggregation of information from neighbors to compose a node representation. Thus, Equation (6) shows how the embedding of a node v given an edge type ζ at the k-th layer, $h_v^{k,\zeta}$ v is computed when the mean aggregator is applied:

$$h_v^{k,\zeta} = \sigma(W_\zeta \cdot MEAN(\{h_v^{k-1,\zeta}\} \cup \{h_u^{k-1,\zeta}, \forall u \in \mathcal{N}_v^\zeta\})) \tag{6}$$

where $\mathcal{N}_v^\zeta$ is the set of neighbors of node v only considering the edges of type ζ and W_ζ are the operator weights considering the edge type ζ. Again, as the input graph $\mathcal{G}_i^{agg}$ comprises 4 different edge types,

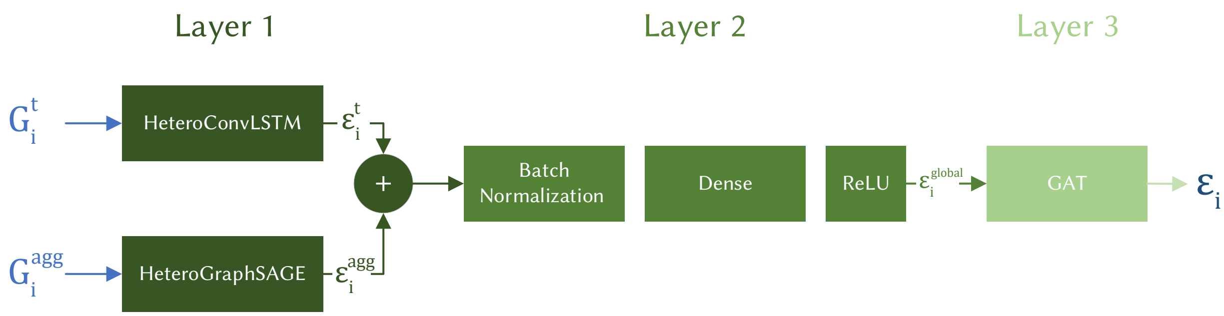


Fig. 11. Layer structure of PRESTO.

the final representation of a node v, $\epsilon_v^{agg} \in \mathcal{E}^{agg}$ is calculated as the aggregation of its embeddings for each type in the last layer, $\varepsilon_v^{agg} = h_v^{(k,collaborate)} + h_v^{(k,creates)} + h_v^{(k,has)} + h_v^{(k,is)}$. It is worth mentioning that the aforementioned computation considers all the neighbors of a node (for a particular edge type) regardless of any temporal constraint. Consequently, the embeddings included in the resulting set $\mathcal{E}_i^{agg}$ provide a complementary view of the time-based embeddings in $\mathcal{E}_i^t$.

Next, the two resulting embeddings from the previous layers are aggregated, normalized and passed through a dense layer with a ReLU activation function, resulting in the embedding set $\mathcal{E}_i^{global}$ (see Fig. 11). By means of this set, we fuse the time-dependent and static views of the nodes for the downstream layers of PRESTO.

Finally, the global embeddings are processed by a Graph ATtention (GAT) layer [37]. This layer incorporates a multi-head attention mechanism that allows weighting the neighbors of a node based on their importance. Hence, Equation (7) refers to how the latent representation of a node v, ϵ_v giving K attention heads, is computed.

$$\varepsilon_v = \sigma(\frac{1}{K}\sum_{k=1}^{K}\sum_{u\in\mathcal{N}_v}\alpha_{vu}^k W^k \varepsilon_u) \tag{7}$$

where α_{vu}^k is the normalization coefficient and W^k is the linear-transformation weight matrix of the *k-th* attention mechanism.

By means of this attention mechanism, the model is able to learn the importance of the links across the four types of edges. For example, it enables giving more importance to the most recent *create* links in case of *non-stationary* artists (whose behaviour is captured by the Conv-LSTM model) or equally-distributed weights across previous *collaborates* edges for *stationary* artists.

Finally, PRESTO returns a matrix $\mathcal{E}_i^{final} \in \mathbb{R}^{|\mathcal{A}|\times\rho}$ in which each artist $a \in \mathcal{A}$ is defined by a ρ-dimensional vector. On the basis of this matrix, we compose a new one $\mathcal{C}_i \in \mathbb{R}^{|\mathcal{A}|\times|\mathcal{A}|}$ with the score for each link between pairs of artists. Equation (8) indicates how such a computation is performed.

$$\mathcal{C}_i = sigmoid(\mathcal{E}_{final}^i \times (\mathcal{E}_{final}^i)^T) \tag{8}$$

This sigmoid function $\mathcal{C}_i$ returns binary labels (0 or 1) for each pair of artists, where $c_{uv} = 1$, $c_{uv} \in \mathcal{C}_i$ indicates that artists u and v are recommended to work together (while 0 means no recommendation). As explained in the next section, this matrix serves as the basis for the final outcome of the recommender system.

C. Generation of the Recommendations

The recommendation procedure of PRESTO can be actually considered an edge prediction problem: Given the time-based and aggregated graphs at the i-th time period, $\mathcal{G}_i^t$ and $\mathcal{G}_i^{agg}$, find a mapping function $\mathcal{P}$ as (9) shows,

$$\mathcal{P}(\mathcal{G}_i^t, \mathcal{G}_i^{agg}) \rightarrow \mathcal{C}_{i+\Delta}^{\delta} \tag{9}$$

where $\mathcal{C}_{i+\Delta}^{\delta}$ is the set of new collaborations among pairs of artists that may occur between the time periods $i+\Delta$ and $i+\Delta+\delta$, being $\Delta \geq 1$ the time horizon and $\delta \geq 0$ the prediction range. The rationale for defining an upper time limit on the prediction outcome $(i+\Delta+\delta)$ is to ensure that the predicted collaborations are not too temporally distant from the input data. Indeed, it would not be very sensible, in operational terms, to predict collaborations that may occur in 5, 10 or 12 years' time.

Consequently, PRESTO should be trained to generate high scores (close to 1) for new collaborations between first-time featured artists within the time slices $i+\Delta$ and $i+\Delta+\delta$. On the other hand, it should generate low scores (close to 0) for links that are unlikely to occur within this time horizon.

V. Evaluation of the Proposal

This section describes the most important results of the evaluation of our proposal following the training goal defined at the end of the previous section.

A. Model Parameters

Table I shows the key parameters of PRESTO used in its evaluation. The last row of this table shows that the GAT layer generated embeddings with 120 features. Hence, the final collaboration matrix $\mathcal{C}_i$ took $|\mathcal{A}| \times 120$ dimensions. We can also see that a *negative* edge sampling rate of 1.0 was used during training. This means that we synthetically generated the same number of negative edges, representing non-existing collaborations, as the number of positive edges reflecting an actual collaboration among artists. Hence, we added to each ground-truth matrix $\mathcal{C}_{i+\Delta}^{\delta}$ the same number of *non-existing* collaborations (labeled as 0) as true ones, thereby enriching the model's learning process.

TABLE I. Model Parameters for the Experiments

Type	Parameter	Value
Training	Loss Train-test split Learning factor Weight decay Optimizer Batch size Neg. edge sampling rate Num. of epochs	Binary Cross Entropy (BCE) 80% train, 20% test 0,001 0.0005 Adam 512 1.0 120
Hetero ConvLSTM, Hetero GraphSAGE	Output size	320
Dense	Input size Output size	320 320
GAT	Num of heads Output size	4 120

TABLE II. F1 Scores Obtained in the Ablation Study for Different Configuration of Time Horizons (Δ) and Prediction Ranges (δ) in Months[a]

δ	6			12			Avg
Δ	12	18	24	12	18	24	
PRESTO	0.765(±0.03)	**0.756** (±0.02)	0.763 (±0.03)	**0.739** (±0.01)	**0.752** (±0.01)	**0.738** (±0.01)	**0.752** (±0.02)
SAGE, Dense, GAT	**0.766** (±0.03)	0.753 (±0.02)	0.760 (±0.07)	0.735 (±0.01)	0.750 (±0.01)	0.737 (±0.01)	0.750 (±0.03)
ConvLSTM, Dense, GAT	0.729 (±0.02)	0.753 (±0.02)	**0.785** (±0.07)	0.723 (±0.01)	0.735 (±0.03)	0.729 (±0.01)	0.742 (±0.03)
SAGE	0.686 (±0.01)	0.704 (±0.03)	0.712 (±0.04)	0.692 (±0.02)	0.691 (±0.03)	0.695 (±0.02)	0.697 (±0.02)
GAT	0.685 (±0.04)	0.707 (±0.05)	0.695 (±0.05)	0.674 (±0.08)	0.701 (±0.06)	0.689 (±0.06)	0.692 (±0.06)
ConvLSTM	0.727 (±0.04)	0.729 (±0.03)	0.732 (±0.03)	0.714 (±0.03)	0.713(±0.04)	0.732 (±0.03)	0.724 (±0.03)

[a] The best score for each ⟨Δ, δ⟩ tuple is shown in bold. The standard deviation is shown in brackets.

TABLE III. ACC Scores Obtained in the Ablation Study for Different Configuration of Time Horizons (Δ) and Prediction Ranges (δ) in Months[a]

δ	6			12			Avg
Δ	12	18	24	12	18	24	
PRESTO	**0.712** (±0.04)	**0.692** (±0.02)	0.715 (±0.04)	**0.674** (±0.02)	**0.683** (±0.02)	**0.670** (±0.02)	**0.692**
SAGE, Dense, GAT	0.709 (±0.04)	0.691 (±0.03)	0.707 (±0.09)	0.667 (±0.02)	0.681 (±0.02)	0.667 (±0.02)	0.688
ConvLSTM, Dense, GAT	0.657 (±0.05)	0.691 (±0.04)	**0.743** (±0.10)	0.644 (±0.02)	0.662 (±0.04)	0.663 (±0.02)	0.677
SAGE	0.584 (±0.04)	0.618 (±0.04)	0.635 (±0.08)	0.604 (±0.04)	0.596 (±0.05)	0.599(±0.02)	0.606
GAT	0.635 (±0.03)	0.640 (±0.06)	0.655 (±0.04)	0.626 (±0.05)	0.638 (±0.05)	0.642 (±0.04)	0.639
ConvLSTM	0.674 (±0.03)	0.669 (±0.04)	0.694 (±0.03)	0.658 (±0.02)	0.658 (±0.03)	0.661 (±0.03)	0.670

[a] The best score for each ⟨Δ, δ⟩ tuple is shown in bold. The standard deviation is shown in brackets.

In implementation terms, PRESTO was developed using the Python 3.8 programming language, along with Pytorch Geometric [38] and Pytorch Geometic Temporal [39] as the primary third-party libraries.

B. Evaluation Metrics

As seen in sec. C, the edge prediction task can be viewed as a binary classification problem. Therefore, we have used the F1 and accuracy (ACC) scores to evaluate the actual accuracy of our model. Equations (10)-(12) shows how the former is calculated:

$$F1 = 2 \times \frac{precision \times recall}{precision + recall} \tag{10}$$

$$recall = \frac{TP}{TP + FN} \tag{11}$$

$$precision = \frac{TP}{TP + FP} \tag{12}$$

where TP, FN and FP refers to True Positives, False Negatives and False Positives, respectively.

Equation (13) shows how the ACC is computed:

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \tag{13}$$

where TN refers to True Negatives.

C. Ablation Study

In order to properly evaluate the suitability of the PRESTO architecture, we conducted an ablation study. Specifically, we compared the F1 score of our proposal with five variations of the model, 1) removing the Conv-LSTM layer (SAGE, Dense, GAT), 2) removing the GraphSAGE (Conv-LSTM, Dense, GAT), 3) only using the GraphSAGE layer, 4) only using the GAT layer and 5) only using the Conv-LSTM layer.

The F1 and ACC results of PRESTO and its variations for different configurations of time horizons Δ and scopes δ are shown in Tables II and III. It is worth noticing that the PRESTO architecture achieved the highest average F1 (0.752) and ACC (0.692) scores, as shown in the rightmost column of these tables.

More in detail, observe that PRESTO clearly outperformed the other alternatives when the time scope δ was set to 12 months. This reflects the model's ability to better predict long-term relationships among artists. For example, our proposal achieved a 0.670 ACC when it predicted future relationships that occurred between 24 and 36 (24+12) months into the future. This score is slightly higher than other alternatives such as the one in which the Conv-LSTM layer, and therefore the temporal processing, was removed (0.667), or the one in which the GraphSAGE layer was not considered (0.663).

Another interesting finding was that the version of the architecture without the Conv-LSTM layer (SAGE, Dense, GAT) provided better results when the time horizon was set to 12 or 18 months. However, the alternative model without the SAGEGraph layer (Conv-LSTM, Dense, GAT) provided better results when the time horizon was set between 18 and 24 months. For example, the former alternative decreased its F1 score from 0.766 to 0.760 when Δ moved from 12 to 24 months with $\delta = 6$, and from 0.750 to 0.737 when Δ changed from 18 to 24 months with $\delta = 12$ (see Table II). On the contrary, (Conv-LSTM, Dense, GAT) increased its F1 score from 0.729 to 0.785 when Δ moved from 12 to 24 months with $\delta = 6$, and it also improved its ACC from 0.644 to 0.667 when Δ moved from 12 to 24 months with $\delta = 12$.

This difference in behavior indicates that the temporal and static parts of the model capture different patterns in the relationships between artists, their songs, and associated genres. The fact that the alternative models, (SAGE, Dense,GAT) and (Conv-LSTM, Dense, GAT) also achieved the best results under certain configurations indicates that neither the Conv-LSTM nor the GraphSAGE layers negatively impact the overall performance of the solution.

Also noteworthy is the fact that the three configurations comprising a single layer obtained significantly lower scores than the other models with at least 3 layers. This is especially noticeable in the models that use only a GraphSAGE or a GAT layer. Indeed, Table III shows that the ACC of both models was 0.585 and 0.635 when $\Delta = 12$ and $\delta = 6$, which are considerably lower than the value obtained by PRESTO (0.712). This reflects the suitability of combining a recurrent layer with an attention mechanism in our proposed approach.

Lastly, Table IV shows the ACC and F1 scores obtained by other recommenders and classifiers within the music industry previously discussed in sec. II. As observed, the PRESTO's values are consistent with the evaluation metrics obtained by other solutions in the literature.

TABLE IV. Comparison With Other Music-Related Classifiers and Recommenders

Proposal	F1	ACC
[24]	-	0.863
[27]	0.880	0.995
[28]	-	0.520
[29]	0.750	-
[30]	-	0.670
PRESTO	0.752	0.692

D. Prediction Examples

For the sake of completeness, Table V shows some predictive outcomes generated by PRESTO during its evaluation. As shown in this table, PRESTO was able to predict collaborations between artists with significantly different levels of popularity such as the remix song *Get Together* by Madonna and Danny Howells (with a prediction horizon of 12 months) or *This is love* by Will.i.am and Eva Simmons. Moreover, it was able to anticipate predictions between musicians with similar popularity, such as the song *Control Myself* by LL. Cool J. and Jennifer Lopez or *Algo Me Gusta de Ti* by Wisin & Yandel and Chris Brown. It is important to note that the system predicted this last collaboration by considering the songs released between 2009-01-01 and 2010-01-01, that is, 24 months before the actual release of the song (2012-01-01).

Regarding the genres of the predicted collaborations, Table V shows examples of Latin tracks (*Amigos del Mundo* and *Bandoleros*), hip-hop and pop tracks (*Live you Life* and *Chasin' Papers*) or soul songs (*Dayglo Reflection* and *American Boy*).

VI. Conclusions and Future Work

Music streaming services, such as Spotify, have become a promising data source for developing new services and business ideas within the music industry. While most of these innovative services are oriented towards end-users (i.e., listeners), such as music recommendation systems, there is also a market niche for offering value-added services to the artists themselves. Particularly, this paper explores how to recommend future collaborations between artists. In this context, collaborations provide opportunities for achieving new commercial success in a competitive music market.

For this goal, we introduced PRESTO, a graph neural network-based system for musical collaboration recommendations. By leveraging data from existing songs created by artists, PRESTO suggests potential collaborations among artists who have not previously worked together. In particular, the system utilizes a heterogeneous graph to model the relationships among artists, songs, music genres, and their temporal evolution. It is trained on 10 song features from two music datasets, namely Spotify and LastFM, including audio properties, contextual data and mood-related attributes of the songs.

The results show an average F1-score of 0.752, demonstrating the system's ability to predict collaborations within a one-year timeframe. Moreover, an ablation study has been conducted to demonstrate that the PRESTO architecture outperforms simpler variations of the same architecture. As a result, PRESTO could be considered an effective tool for providing valuable recommendations for artists and music companies, facilitating the production of new tracks and supporting artists in their career development.

There are several directions to further improve and expand PRESTO. Firstly, we plan to incorporate additional data sources, such as the artists' social media activity. These data could provide a more comprehensive understanding of artists' careers and music styles, leading to better recommendations. A second line of future research will involve integrating listener feedback on the collaborations suggested by PRESTO to enhance the performance of the system over time. Lastly, evaluating the real-world impact of PRESTO by measuring the outcomes of the recommended collaborations, such as streaming numbers, user engagement, and artist success, could provide valuable insights for the music industry.

Acknowledgment

Financial support for this research has been provided under grant PID2020-112827GB-I00 funded by MCIN/AEI/10.13039/501100011033.

Code Availability

The source code of PRESTO is available at https://github.com/fterroso/the_presto_app

References

[1] P. Williamson, J. Luebbers, "Expanding models of music composition: Exploring the value of collaboration," *International Journal of Music Education*, vol. 41, no. 1, pp. 111–128, 2023, doi: 10.1177/02557614221090520.

[2] International Federation of the Phonographic Industry (IFPI), "Global music report 2023: State of the industry," 2023. [Online]. Available: https://www. ifpi.org/wp-content/uploads/2020/03/Global_ Music_ Report_2023_State_of_the_Industry.pdf, Accessed: Jun 29th, 2025.

[3] J. Parc, Y. Kim, "Analyzing the reasons for the global popularity of BTS: A new approach from a business perspective," *Journal of International Business and Economy*, vol. 21, pp. 15–36, Nov. 2020, doi: 10.51240/jibe.2020.1.2.

[4] R. D. Smith, "The network of collaboration among rappers and its community structure," *Journal of Statistical Mechanics: Theory and Experiment*, p. P02006, feb 2006, doi: 10.1088/1742-5468/2006/02/P02006.

[5] Spotify, "The crossover effect: Artist collaborations thrive on Spotify," 2023. [Online]. Available: https://newsroom.spotify.com/2023-05-24/crossover- collaborations-genre-collabs-streaming-data-spotify/, Accessed: May 15th, 2024.

[6] C. McDuffie, "Comprender las estadísticas de los fans: contribuciones de los ejecutivos de luminate," 2022. [Online]. Available: https://artists.spotify.com/es-419/blog/ understanding-fan-analytics-luminate-execs-weigh-in, Accessed: May 15th, 2024.

[7] M. O. Silva, G. P. Oliveira, D. B. Seufitelli, M. M. Moro, "Temporal success analyses in music collaboration networks: Brazilian and global scenarios," *Revista Vórtex*, vol. 11, pp. 1–27, Aug. 2023, doi: 10.33871/23179937.2023.11.2.7185.

[8] J. McKenzie, P. Crosby, L. J. Lenten, "It takes two, baby! feature artist collaborations and streaming demand for music," *Journal of Cultural Economics*, vol. 45, pp. 385– 408, 2021.

[9] Q. Dai, X.-M. Wu, L. Fan, Q. Li, H. Liu, X. Zhang, D. Wang, G. Lin, K. Yang, "Personalized knowledge- aware recommendation with collaborative and attentive graph convolutional networks," *Pattern Recognition*, vol. 128, p. 108628, 2022, doi: https://doi.org/10.1016/j.patcog.2022.108628.

[10] D. Holtz, B. Carterette, P. Chandar, Z. Nazari, H. Cramer, S. Aral, "The Engagement-Diversity Connection: Evidence from a Field Experiment on Spotify," in *Proceedings of the 21st ACM Conference on Economics and Computation*, EC '20, Virtual Event, Hungary, 2020, pp. 75–76.

[11] I. Anderson, S. Gil, C. Gibson, S. Wolf, W. Shapiro, O. Semerci, D. M. Greenberg, ""Just the Way You Are": Linking Music Listening on Spotify and Personality," *Social Psychological and Personality Science*, vol. 12, no. 4, pp. 561–572, 2021, doi: 10.1177/1948550620923228.

[12] M. O. Silva, L. M. Rocha, M. M. Moro, "Collaboration profiles and their impact on musical success," in *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*, SAC '19, New York, NY, USA, 2019, pp. 2070–2077, Association for Computing Machinery.

[13] F. Calefato, G. Iaffaldano, L. Trisolini, F. Lanubile, "An in-depth analysis of occasional and recurring collaborations in online music co-creation," *Trans. Soc. Comput.*, vol. 4, Jan 2022, doi: 10.1145/3493800.

[14] A. H. Raza, K. Nanath, "Predicting a hit song with machine learning: Is there an apriori secret formula?," in *2020 International Conference on Data Science, Artificial Intelligence, and Business Analytics (DATABIA)*, 2020, pp. 111–116, IEEE.

[15] D. Paul, S. Kundu, "A survey of music recommendation systems with a proposed music recommendation system," in *Emerging Technology in Modelling and Graphics: Proceedings of IEM Graph 2018*, Kolkata, India, 2020, pp. 279–285, Springer.

[16] G. Björklund, M. Bohlin, E. Olander, J. Jansson, C. E. Walter, M. Au-Yong-Oliveira, "An exploratory study on the Spotify recommender system," in *Information Systems and Technologies: WorldCIST 2022, Volume 2*, Budva, Montenegro: Springer, 2022, pp. 366–378.

[17] A. Werner, "Organizing music, organizing gender: algorithmic culture and spotify recommendations," *Popular Communication*, vol. 18, no. 1, pp. 78–90, 2020.

[18] M. Schedl, "Deep learning in music recommendation systems," *Frontiers in Applied Mathematics and Statistics*, vol. 5:44, 2019.

[19] V. La Gatta, V. Moscato, M. Pennone, M. Postiglione, G. Sperlí, "Music recommendation via hypergraph embedding," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 10, pp. 7887–7899, 2023, doi: 10.1109/TNNLS.2022.3146968.

[20] H. Weng, J. Chen, D. Wang, X. Zhang, D. Yu, "Graph-based attentive sequential model with metadata for music recommendation," *IEEE Access*, vol. 10, pp. 108226–108240, 2022, doi: 10.1109/ACCESS.2022.3213812.

[21] H.-G. Kim, G. Y. Kim, J. Y. Kim, "Music recommendation system using human activity recognition from accelerometer data," vol. 65, p. 349–358, Aug 2019, doi: 10.1109/TCE.2019.2924177.

[22] F. Fessahaye, L. Perez, T. Zhan, R. Zhang, C. Fossier, R. Markarian, C. Chiu, J. Zhan, L. Gewali, P. Oh, "T- recsys: A novel music recommendation system using deep learning," in *2019 IEEE international conference on consumer electronics (ICCE)*, 2019, pp. 1–6, IEEE.

[23] J. P. Singh, J. K. Bhatia, "Scientific exploration of music recommendation system using facial expressions," in *Advances in Data and Information Sciences: Proceedings of ICDIS 2022*, Leipzig, Germany, 2022, pp. 11–21, Springer.

[24] X. Wen, "Using deep learning approach and iot architecture to build the intelligent music recommendation system," *Soft Computing*, vol. 25, pp. 3087–3096, 2021.

[25] T. Li, M. Choi, K. Fu, L. Lin, "Music sequence prediction with mixture hidden markov models," in *2019 IEEE International Conference on Big Data (Big Data)*, 2019, pp. 6128–6132, IEEE.

[26] F. Pachet, "Hit song science," in *Music data mining, T. Li and M. Ogihara and G. Tzanetakis Eds.*, 2011, pp. 305–326, Chapman & Hall/CRC Press Boca Raton, FL, USA.

[27] R. Rajyashree, A. Anand, Y. Soni, H. Mahajan, "Predicting Hit Music using MIDI features and Machine Learning," in *2018 3rd International Conference on Communication and Electronics Systems (ICCES)*, 2018, pp. 94–98.

[28] A. H. Raza, K. Nanath, "Predicting a Hit Song with Machine Learning: Is there an apriori secret formula?," in *2020 International Conference on Data Science, Artificial Intelligence, and Business Analytics (DATABIA)*, 2020, pp. 111–116.

[29] E. Zangerle, M. Vötter, R. Huber, Y.-H. Yang, "Hit song prediction: Leveraging low-and high-level audio features.," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Delf, Netherlands, 2019, pp. 319–326.

[30] M. Mayerl, M. Vötter, G. Specht, E. Zangerle, "Pairwise learning to rank for hit song prediction," in *Proceedings of BTW 2023*, Dresden, Germany, 2023, pp. 555–565.

[31] M. Schedl, S. Brandl, O. Lesota, E. Parada-Cabaleiro, D. Penz, N. Rekabsaz, "Lfm-2b: A dataset of enriched music listening events for recommender systems research and fairness analysis," in *ACM SIGIR Conference on Human Information Interaction and Retrieval*, CHIIR '22, New York, NY, USA, 2022, p. 337–341, Association for Computing Machinery.

[32] M. d. O. Santos-Silva, *Collaboration-aware hit song analysis and prediction.* PhD dissertation, Universidade Federal de Minas Gerais, 2020.

[33] C. E. Shannon, "A mathematical theory of communication," *The Bell system technical journal*, vol. 27, no. 3, pp. 379–423, 1948.

[34] Y.-W. Cheung, K. S. Lai, "Lag order and critical values of the augmented dickey–fuller test," *Journal of Business & Economic Statistics*, vol. 13, no. 3, pp. 277–280, 1995.

[35] Z. Yuan, X. Zhou, T. Yang, "Hetero-convlstm: A deep learning approach to traffic accident prediction on heterogeneous spatio-temporal data," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '18, New York, NY, USA, 2018, pp. 984–992, Association for Computing Machinery.

[36] W. Hamilton, Z. Ying, J. Leskovec, "Inductive representation learning on large graphs," *Advances in neural information processing systems*, vol. 30, 2017.

[37] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio, Y. Bengio, *et al.*, "Graph attention networks," *stat*, vol. 1050, no. 20, pp. 10–48550, 2017.

[38] M. Fey, J. E. Lenssen, "Fast graph representation learning with PyTorch Geometric," in *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.

[39] B. Rozemberczki, P. Scherer, Y. He, G. Panagopoulos, A. Riedel, M. Astefanoaei, O. Kiss, F. Beres, G. Lopez, N. Collignon, R. Sarkar, "PyTorch Geometric Temporal: Spatiotemporal Signal Processing with Neural Machine Learning Models," in *Proceedings of the 30th ACM International Conference on Information and Knowledge Management*, 2021, p. 4564–4573.

Fernando Terroso-Sáenz

He obtained his B.S. and PhD in Computer Science from the University of Murcia in 2009 and 2013. Since 2023 he is associate professor at Technical University of Cartagena (UPCT). He has published more than 50 articles in international journals and congresses. His research areas include smart mobility, human-generated data analysis and mobile sensing.

Jesús Soto

He received the M.Sc. degree in mathematics from University of Murcia in 1993 and the Ph.D. degree in applied mathematics from University of Alicante in 2005. He is the author of 22 research articles, as well as national and international conferences. He has participated in several regional and national projects, such as PROMETEO, SENECA and "Retos-Colaboración". His research interests include time series analysis and data mining, in particular, fuzzy clustering.

Andrés Muñoz

He is an associate professor in the University of Cadiz. He received his B.S. degree in Computer Science (2005) and PhD degree in Computer Science (2011) from University of Murcia (Spain). His main research interests include Semantic Web technologies, Ambient Intelligence and Intelligent Environments.

Philippe Roose

He is Full Professor at LIUPPA research lab – E2S, University of Pau et des Pays de l'Adour (France). He is co-inventor of both KalimuchoTM & PISCO platforms (4 Int'l patents). He supervised more than 20 PhD thesis (3 current) and was involved in more than 40 PhD defenses. He published many national and international articles, journals and books and organized several national and international conferences as well. His works are mainly focuses on green computing, middleware, software architecture, context- aware and autonomic.