Schedule-based Harmonic Playlist Generation using Neural Networks

Rym Benchaabane
Electrical Engineering Department
University of Houston
Houston, TX
benchaabane.rym@gmail.com

Rhema Ike
Electrical Engineering Department
University of Houston
Houston, TX
rhemaike@gmail.com

Javier Garcia Gonzalez
Electrical Engineering Department
University of Houston
Houston, TX
jgarciagonzalez@uh.edu

***Abstract*—In recent years, the goal of an automated playlist generation has been a common interest. While music listeners can choose the songs they like and arrange them manually based on their mood and foreseeable activities, several platforms such as Spotify and YouTube have options to automatically generate these playlists based on some patterns, usually liked genres or artists. In this paper, we take a step further and implement a machine learning method to create a playlist tailored to a listener's schedule, taking into account their daily activities. In addition, the playlist produced by this method attempts to follow the Camelot system, a tool intended for harmonic mixing of songs.**

The proposed method consists of three major tasks. First, every song in a collection is given a label among four possible activities. The assigned label is based on 11 song characteristics. Afterwards, the schedule of the listener is analyzed in time steps to determine how the playlist should be structured. This is done by taking into account the order of activities as well as the time remaining in the current activity for every time step. Finally, compatible songs are added to the playlist to satisfy the required structure. Song compatibilty is dictated by the Camelot system. The first two tasks utilize Feedforward Neural Networks (NNs), while the last one utilizes two Recurrent Neural Networks (RNNs).

***Index Terms*—music, playlist, harmonic mixing, machine learning, neural network**

A	Perfect match	Energy boost +	++	+++	Energy drop -	- -	- - -	Mood change
1A	1A, 12B	1B, 2A	10A	3A, (8A)	12A	4A	11A, (6A)	4B
2A	2A, 1B	2B, 3A	11A	4A, (9A)	1A	5A	12A, (7A)	5B
3A	3A, 2B	3B, 4A	12A	5A, (10A)	2A	6A	1A, (8A)	6B
4A	4A, 3B	4B, 5A	1A	6A, (11A)	3A	7A	2A, (9A)	7B
5A	5A, 4B	5B, 6A	2A	7A, (12A)	4A	8A	3A, (10A)	8B
6A	6A, 5B	6B, 7A	3A	8A, (1A)	5A	9A	4A, (11A)	9B
7A	7A, 6B	7B, 8A	4A	9A, (2A)	6A	10A	5A, (12A)	10B
8A	8A, 7B	8B, 9A	5A	10A, (3A)	7A	11A	6A, (1A)	11B
9A	9A, 8B	9B, 10A	6A	11A, (4A)	8A	12A	7A, (2A)	12B
10A	10A, 9B	10B, 11A	7A	12A, (5A)	9A	1A	8A, (3A)	1B
11A	11A, 10B	11B, 12A	8A	1A, (6A)	10A	2A	9A, (4A)	2B
12A	12A, 11B	12B, 1A	9A	2A, (7A)	11A	3A	10A, (5A)	3B
B	Perfect match	Energy boost +	++	+++	Energy drop -	- -	- - -	Mood change
1B	1B, 2A	2B	10B	3B, (8B)	1A, 12B	4B	11B, (6B)	10A
2B	2B, 3A	3B	11B	4B, (9B)	2A, 1B	5B	12B, (7B)	11A
3B	3B, 4A	4B	12B	5B, (10B)	3A, 2B	6B	1B, (8B)	12A
4B	4B, 5A	5B	1B	6B, (11B)	4A, 3B	7B	2B, (9B)	1A
5B	5B, 6A	6B	2B	7B, (12B)	5A, 4B	8B	3B, (10B)	2A
6B	6B, 7A	7B	3B	8B, (1B)	6A, 5B	9B	4B, (11B)	3A
7B	7B, 8A	8B	4B	9B, (2B)	7A, 6B	10B	5B, (12B)	4A
8B	8B, 9A	9B	5B	10B, (3B)	8A, 7B	11B	6B, (1B)	5A
9B	9B, 10A	10B	6B	11B, (4B)	9A, 8B	12B	7B, (2B)	6A
10B	10B, 11A	11B	7B	12B, (5B)	10A, 9B	1B	8B, (3B)	7A
11B	11B, 12A	12B	8B	1B, (6B)	11A, 10B	2B	9B, (4B)	8A
12B	12B, 1A	1B	9B	2B, (7B)	12A, 11B	3B	10B, (5B)	9A

Fig. 1. Table representation of the Camelot system. Each of the 24 combinations (12 keys with two modes each) produces different results when mixed with the combinations in their rows. Changes in energy are associated with the excitement level of the listener.

I. Introduction

Music is a big part of many people's lives today. Whether to help relax, get over difficult times, or increase productivity at work, listening to a playlist that has the right type of songs can make a difference in satisfying the need at hand. As examples, research results from [4], [2] and [3] show that lively music, in intervals of around 10 minutes help boost productivity when performing repetitive tasks, and that song features such as tempo and modality can have an impact on quality of communication.

Some individuals have the time and dedication to curate large collections of songs and to prepare playlists in advance for their activities. Others rely on platforms that automatically generate these playlists based on a listener's taste and need. These popular platforms are popular because they provide an easier and faster service. One platform that is particularly popular is Spotify, which has become the leading music streaming service in recent years with over 270 million reported users worldwide ([1]). Spotify provides the listener with different automatically generated playlists, for different purposes such as discovering songs or keeping track of new releases. These playlists are created by taking into account the listener's most liked songs and genres and finding new tracks that match said preferences.

In truth, how suitable a playlist is for a person at a particular time is not only dependent on what their general listening habits are but also what their current situation is. Factors such as activity performed and current mood impact the desire of listening to a particular sequence of songs, and automatically creating playlists taking into account such factors is more complex than simply relying on listening history. Research has already been conducted on how to generate more complex sequences when using more information about the listener as input, as it is discussed in Section II.

In addition to selecting a song that is compatible with the current state of the listener, it is important to take care that the transition between songs is of an acceptable nature. While a track might be suited to continue a playlist, it might result in an undesirable change with respect to the previous song. Music theory has developed multiple tools to help create a good sequence in this regard, and we choose to follow the Camelot system in order to force song transitions to be pleasant and natural. The Camelot system is a simplified version of the Circle of Fifths, a visual representation of the relationship of musical keys, and describes the relationship between the keys of two songs [10]. Therefore, we can determine if two songs are perfectly matched, if the combination results in a positive or negative energy change, which we refer to as harmonic mixing, or if they clash and therefore decrease the quality of the playlist.

II. Related Work

Several approaches to automatically create complex playlists have been studied and evaluated. The common trend is utilizing multiple inputs related to the listener when choosing and building a song sequence.

A straightforward improvement is to look at listener and song related data in several platforms. Germain and Chakareski consider the listener's history in Spotify and the ranking (number of likes) of artists' Facebook pages [5]. Input vectors contain the most liked artists (F_b), and the unique artists (S_{p1}) and the most recently played songs (S_{p2}) in a listener's Spotify library. F_b, S_{p1} and S_{p2} are used together with EchoNest to find recommendations. Utilizing the WTF score (fraction of disliked songs and novelty factor fraction of new liked songs), experiments with ten different subjects showed an improvement of four percent over Spotify radio when it came to the ratio of liked to disliked music recommendations. Another approach is discussed by Bohra et al. in [6], where a bank of songs is first fragmented into several segments related to different activities, with some songs remaining uncategorized. The playlist then chooses random tracks from appropriate segments based on the current activity of the listener. Results showed that after 100 days of testing, the number of skipped songs in a playlist resulting from this method approaches zero while the number of skips in a randomly generated playlist remained at around 200.

Continuing the trend on the current context of the listener, Pauws and Egen take into account how particular songs fit the context for which the playlist is being generated, focusing more on the listener's recent musical preference rather than their overall musical tastes [7]. Jazz songs were given labels for 18 attributes, ranging from style and tempo to number of musicians and recording place. When a listener chooses the starting song, a local cluster method finds songs in the vicinity, in order to choose similar yet varied entries. An inductive learning algorithm then constructs a decision tree based on the user's feedback, and categorizes songs into preferred and rejected options for the playlist. Participants were asked to rate the results for two different contexts, soft music and live music, resulting in a mean rating score close to 7.5 for both cases versus mean rating scores of less than 6.0 for randomly generated sequences.

In [8], Oliver and Kreger-Stickles increase the complexity of the task by having the playlist help the listener achieve a certain task at hand. On top of being context-aware like in the previous work, the listener's physiological response is continuously monitored to evaluate if the playlist is having a desired effect. If the listener has relaxing as a goal, then the algorithm will choose more relaxing songs until the physiological response of the listener changes accordingly. In this particular application the playlist is designed to serve as a trainer of sorts, encouraging the listener to run at a certain pace based on a heart-rate goal. Results for two runners indicated they had a workout closer to intended than without music or with random playlists.

Irene et al. focuses on the structure of a playlist [9]. A set of target characteristics is observed in order to select candidate songs to continue an existing sequence. Unlike [8], the target is not a particular physiological response from the user, but maintaining a cohesive relationship, or trend, between songs with respect to features such as rhythm or release date. A two-layer Recurrent Neural Network (RNN) with Long Short-Term Memory (LSTM) cells is employed to model the evolution of sample sequences. New and varied playlists can be created in this manner since the starting song can change, and the ideal features requested by the network might not perfectly fit a specific song. The results show a mean absolute error of less than five percent when the generated playlists are compared to the respective desired counterparts.

For our work we take much inspiration from [9], with the goal of generating playlists with target structures. Mentioned structures are not only related to the Camelot system but also to the set of activities, or schedule, of a particular listener. This adds novelty to our approach because the listener's context directs the evolving trend of the playlist, and also differentiates our work from that of [6] and [7] in that we consider a whole schedule when generating a sequence, as opposed to a single activity. However, we do apply some of the techniques discussed in those two works to segment our song bank into different categories based on several features, before running our algorithms to create a playlist.

III. Methods

The proposed method is comprised of three main parts or tasks. First, all the available songs are classified into one or more activities based on several features. Second, the state of a listener's schedule is used to determine if a playlist needs an increase or decrease in energy as defined by the Camelot system. Finally, one of two independently trained RNNs tests candidate songs to check if they result in harmonic mixing when added to the playlist. A visualization of the general process is shown in Fig 2. The code was developed in Python with the help of the TensorFlow library.

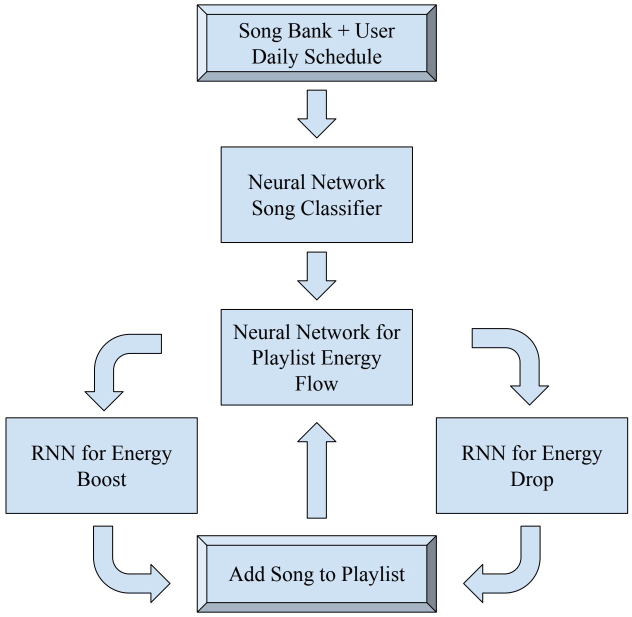

Fig. 2. Overview of the playlist generation method. The songs are classified into activity categories, and the schedule of the listener is used to determine the energy flow of the playlist. Based on the flow requirement, one of two RRNs adds a song to the playlist.

A. Song acquisition and classification

The song data for our implementation was acquired through the Spotify API. In total, 3200 songs with 11 features each were used to train a NN to categorize songs into one of four activities: beastmode gym, beach party, studying and relaxing. Before training the model the features were normalized with the exception of the mode which can only take one of two values (0 for A and 1 for B). An example of a song and its relevant features is shown in Table I. The activity labels are assigned subjectively in this implementation, as in practice a listener can assign them to songs in their collection according to their own preference.

TABLE I
SONG FEATURES

Name	Key	Mode	Liveness
”Hidden Road”	1	1	0.101
Instrumentalness	**Loudness**	**Speechiness**	**Tempo**
0.871	0.537	0.035	0.424
Valence	**Danceability**	**Energy**	**Acousticness**
0.325	0.331	0.111	0.991

B. Playlist Energy Flow

With the songs classified, the next step is to train a second NN to determine whether the playlist should have an increase or decrease in energy. It is important to note that this energy is not the same as the song feature of the same name, but the concept described by the Camelot system. The decision

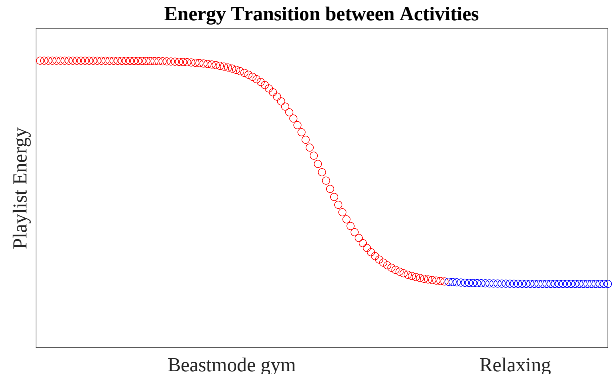

Fig. 3. Example of the change in energy of a playlist as activities change. In this case, the energy begins to decrease at the end of the beast mode gym time period because the next activity, relaxing, requires calmer music.

is made based on the current activity being performed, the time left in said activity, and the next activity in the schedule if applicable. As an example, if a listener is currently in beast mode gym with less than half of the time allotted remaining, and the next activity is relaxing, the NN will start recommending songs that would decrease the energy level of the playlist (Figure 3). The goal is to have a smoother transition between activities, helping the listener progressively get motivated or relaxed in preparation for what is to come, than what would result if the algorithm simply switched from randomly picking songs from one activity category to the other when the transition happens.

This NN does not discriminate between the three possible levels of energy boosting or dropping for simplicity purposes. Its output is simply whether there should be an energy boost, energy decrease, or no change (perfect match). Additionally, the mood change aspect of the Camelot system is not considered as an output option. Over 3000 examples of activity transitions were used to train this NN.

C. Camelot Wheel Implementation for Energy Changes

While the NN in the previous step determines how the energy of the playlist should change as the schedule progresses, the keys of the songs in the sequence should follow the Camelot system to result in harmonic mixing. Two different RNNs are trained to this end, one for energy boosts and the other for energy drops. Considering a playlist $P = S_1, S_2, ..., S_{n-1}$, the RNNs evaluate whether a song S_n is a good follow up to S_{n-1} based on their key and activity. The RNNs were trained with 750 S_{n-1}, S_n sequence examples containing the aforementioned features for both songs. (need to work on this, going to sleep doe). If a perfect match is needed, the algorithm just cycles through songs in the current activity category until it finds one with the required key.

IV. RESULTS

Once training was done, 800 songs were used to validate the song classification model, about 800 activity transitions were

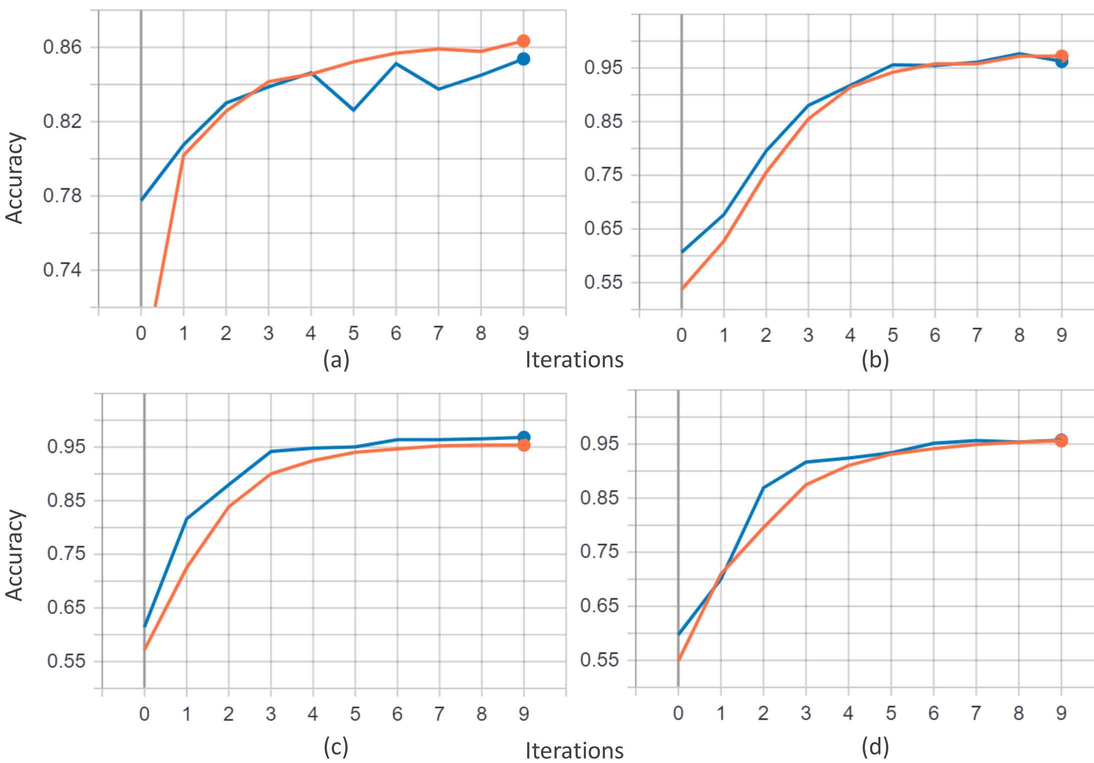

Fig. 4. Graphs of the training (orange) and validation (blue) accuracies versus iteration for (a) the song classifying NN, (b) the energy flow NN, (c) the RNN for energy boosting and (d) the RNN for energy dropping. Each iterations runs the appropriate data set.

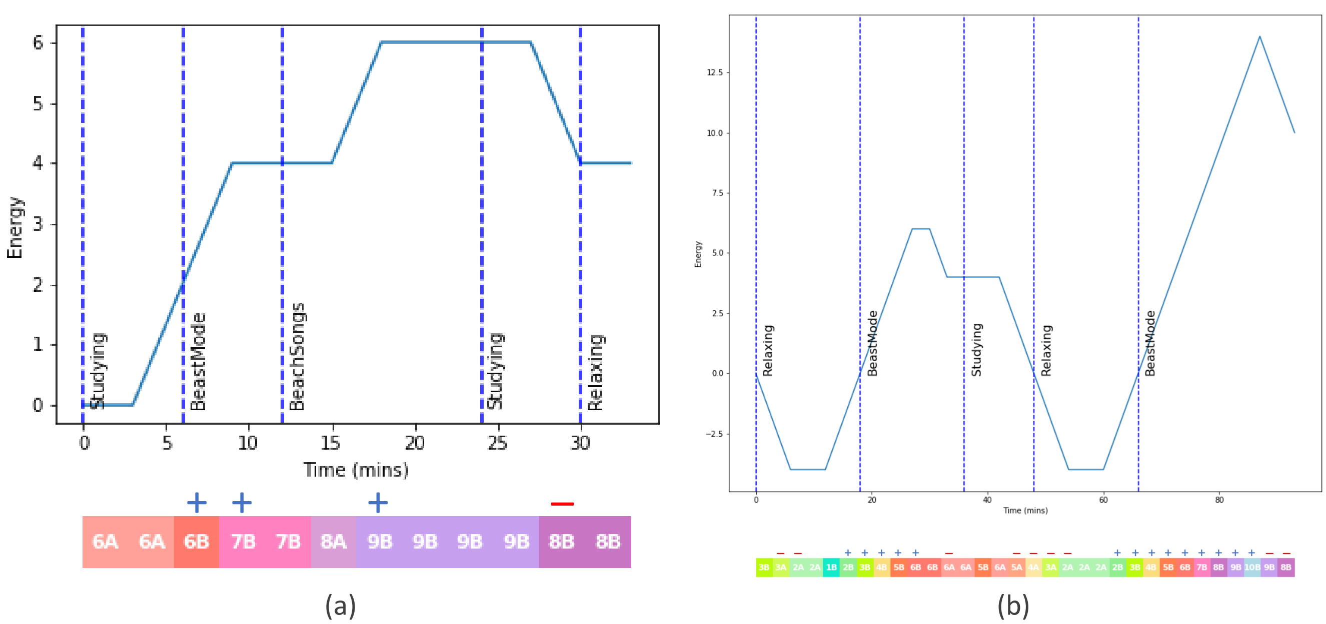

Fig. 5. Generated playlists for two different schedules. Both schedules consist of five activities, but playlist (a) is considerably shorter in time and only has one major shift in energy change, while (b) incorporates several changes. The keys of the individual songs that make up the playlists are shown beneath the respective graph, and ther

used to validate the energy flow model, and 4000 sequences were used for each RNN. These values correspond to 20%, 20% and 25% of our entire data sets respectively.

The training and validation accuracies of the song classification and energy flow NNs, as well as those of the sequence RNNs, and are shown in Figure 4. The training accuracy approaches an upper limit as is desired. The song classification NN resulted in an accuracy of 86.4%, while the energy flow NN had a final accuracy of 97.1%. The song classification model has 8 more features than the energy flow model, a likely reason for the latter having better training results. The RNNs also had high final accuracies of 96% and 95.6%.

The validation accuracies follow the trend of their training counterparts. The song classification NN had a final accuracy of 85.7%, and the energy flow NN had 95.9%. The RNNs finished validation with 96.3% and 95.7% accuracy. It is reasonable to declare that our models did not suffer from over-fitting, based on these results.

Figure 5 shows two examples of a schedule and the resulting playlist energy. For each playlist there is a graph showing its energy as a function of time, with vertical lines indicating the activities in the schedule. Since the Camelot system only describes the change in energy, not actually giving numeric values to specific energy levels, the value at the beginning is always 0.

For the first example, the listener's schedule starts with studying and has beast mode gym as the second activity. Accordingly, the energy of the playlist starts increasing as the transition approaches. During the beast mode gym and beach party activities the energy keeps increasing, as the goal is to keep the listener excited. The last two activities, studying and relaxing, are relatively calm and so the energy drops towards the end of the schedule.

The second example follows the same transition trends as the first one. One difference is that, unlike the first schedule which resulted in a steady energy increase followed by a steady decrease, this schedule has a relaxing activity in between two beast mode gym ones. Its playlist correctly adapts by dropping the energy, staying constant for a bit in the middle of the relaxing activity, and rising it up again.

The keys of the individual songs that make up the playlists are also shown in Figure 5, underneath the energy graphs. Plus and minus signs indicate energy boosts and drops respectively. Since the keys chosen to generate the changes agree with the Camelot system, adjacent songs in the playlists theoretically mix with each other in a harmonious manner.

V. Conclusion

The proposed playlist generation method accurately groups songs in a way that matches a listener's schedule, following the rules in the Camelot system to harmoniously boost or drop the energy of the playlist according to the activities to be performed. By first classifying the available songs into the activity categories and training a model that dictates the change in energy required for activity transitions, our RNNs were able to create sequences that satisfied extended schedule needs.

While the songs, song labels and activity transitions used in this implementation were provided by us, and thus are subjective, each listener would provide their own information for those categories in a hypothetical commercial version. The only model that is universal are the sequence RNNs, since the Camelot system is not based on musical preference.

For this paper only the objective characteristics of the method were evaluated and presented, namely accuracy of the neural networks and the key progression of generated playlists. There was no time to have volunteers evaluate the playlist, and give it a subjective score in a similar fashion to [5] and [7].

For future work, there is the possibility of increasing the complexity of the energy flow model by implementing different levels for energy boosting and dropping in the Camelot System. The tempo is also another important aspect to take into consideration while transitioning. The outputs of the model can also be expanded to include a mood change option. While this method requires listeners to have previously provided data for the models to be trained, feedback such as that implemented in [8] can be implemented so the models can learn from the listener's reaction to the generated playlists.

References

[1] Aguiar, Luis and Waldfogel, Joel, "Platforms, Promotion, and Product Discovery: Evidence from Spotify Playlists",National Bureau of Economic Research Working Paper Series, No. 24713, 2018

[2] J. G. Fox and E. D. Embrey, "Music - an aid to productivity," Applied Ergonomics, 3(4), 1972, pp. 202–205.

[3] D. J. Blood and S. J. Ferriss, "Effects of Background Music on Anxiety, Satisfaction with Communication, and Productivity," Psychological Reports, 72(1), 1993, pp. 171–177.

[4] 11. C. Sun et al. "The effect of different types of music on electroencephalogram" 2013 IEEE International Conference on Bioinformatics and Biomedicine pp. 31-37 2013.

[5] A. Germain and J. Chakareski, "Spotify Me: Facebook-assisted automatic playlist generation," 2013 IEEE 15th International Workshop on Multimedia Signal Processing (MMSP), Pula, 2013, pp. 025-028.

[6] T. S. Bohra, V. Kumar and S. Ganesan, "Segmenting music library for generation of playlist using machine learning," 2015 IEEE International Conference on Electro/Information Technology (EIT), Dekalb, IL, 2015, pp. 421-425.

[7] S. Pauws and B. Eggen, "PATS: Realization and User Evaluation of an Automatic Playlist Generator," 3rd International Conference on Music Information Retrieval (ISMIR), Paris, France, 2002.

[8] N. Oliver and L. Kreger-Stickles, "PAPA: Physiology And Purpose-Aware Automatic Playlist Generation," 7th International Conference on Music Information Retrieval (ISMIR) Victoria, Canada, 2006, pp. 250-253.

[9] R. T. Irene, C. Borrelli, M. Zanoni, M. Buccoli and A. Sarti, "Automatic playlist generation using Convolutional Neural Networks and Recurrent Neural Networks," 2019 27th European Signal Processing Conference (EUSIPCO), A Coruna, Spain, 2019, pp. 1-5.

[10] Parker, Joseph. "Camelot Wheel Tricks to Advance Your DJ Skills." Best DJ Gear, www.bestdjgear.net/camelot-wheel-tricks-to-advance-your-dj-skills/.