applied sciences

MDPI

Article

An Emotion-Aware Personalized Music Recommendation System Using a Convolutional Neural Networks Approach

Ashu Abdul [1], Jenhui Chen [2,3,4,*,†], Hua-Yuan Liao [2] and Shun-Hao Chang [2]

[1] Department of Electrical Engineering, College of Engineering, Chang Gung University, Kweishan, Taoyuan 33302, Taiwan; ashu.a507@gmail.com

[2] Department of Computer Science and Information Engineering, College of Engineering, Chang Gung University, Kweishan, Taoyuan 33302, Taiwan; nmxzqw@gmail.com (H.-Y.L.); abide0222@gmail.com (S.-H.C.)

[3] Department of Otorhinolaryngology, Head and Neck Surgery, Chang Gung Memorial Hospital, Kweishan, Taoyuan 33375, Taiwan

[4] Department of Electronic Engineering, Ming Chi University of Technology, Taishan Dist., Taipei 24301, Taiwan

* Correspondence: jhchen@mail.cgu.edu.tw; Tel.: +886-3-211-8800 (ext. 5990)

† This work was supported in part by the Ministry of Science and Technology, Taiwan, under Contract MOST 106-2221-E-182-065.

Received: 31 May 2018; Accepted: 6 July 2018; Published: 8 July 2018

Abstract: Recommending music based on a user's music preference is a way to improve user listening experience. Finding the correlation between the user data (e.g., location, time of the day, music listening history, emotion, etc.) and the music is a challenging task. In this paper, we propose an emotion-aware personalized music recommendation system (EPMRS) to extract the correlation between the user data and the music. To achieve this correlation, we combine the outputs of two approaches: the deep convolutional neural networks (DCNN) approach and the weighted feature extraction (WFE) approach. The DCNN approach is used to extract the latent features from music data (e.g., audio signals and corresponding metadata) for classification. In the WFE approach, we generate the implicit user rating for music to extract the correlation between the user data and the music data. In the WFE approach, we use the term-frequency and inverse document frequency (TF-IDF) approach to generate the implicit user ratings for the music. Later, the EPMRS recommends songs to the user based on calculated implicit user rating for the music. We use the million songs dataset (MSD) to train the EPMRS. For performance comparison, we take the content similarity music recommendation system (CSMRS) as well as the personalized music recommendation system based on electroencephalography feedback (PMRSE) as the baseline systems. Experimental results show that the EPMRS produces better accuracy of music recommendations than the CSMRS and the PMRSE. Moreover, we build the Android and iOS APPs to get realistic data of user experience on the EPMRS. The collected feedback from anonymous users also show that the EPMRS sufficiently reflect their preference on music.

Keywords: convolutional neural networks; latent features; machine learning; music; user preference; weighted feature extraction

1. Introduction

Personalized music recommendation approaches are used by many online music stores and streaming services (e.g., iTunes (https://www.apple.com/itunes/download/), Spotify (https://www.spotify.com/, KKBox (https://www.kkbox.com/tw/tc/index.html), Grooveshark (http://groovesharks.org/), etc.) to understand users' music preference [1]. These approaches learn the user's preference by analyzing the user's music listening history for providing music recommendations to a user. For simplicity, we will use the term 'the song' to represent 'the music' listened by the user through out this paper. The music recommendation to the user is the list of songs recommended to the user. There are three main approaches for personalized music recommendations: the content-based (CB) [2], the collaborative filtering (CF) [3], and the hybrid approach [4]. The CB recommendations approach recommends similar songs to the user based on songs presented in the user's music listening history. The user's music listening history represents the previously listened songs by the user. The CF recommendation approach recommends songs to a user based on songs listened by the group of people who have similar preferences to that of the user. The hybrid approach incorporates the knowledge obtained from the CB and the CF approaches for recommending songs to the user.

Alajanki, Wiering, and Veltkamp [5] proposed a system to extract the user's preference from the user's music listening history. Based on the user's preferences, the system recommends songs to the user. Their system was limited to extracting the user's preference on the user's music listening history. However, their system does not concentrate on how to extract the user's preference based on the user's information (such as age, gender, location, ambient, time of the data, emotions etc.). In this paper, we consider three user emotions: happy, normal and sad. For example, we may feel relaxed when listening to the songs from a romantic movie or we may feel sad when listening to the songs from a horror movie. Shan et al. [6] tried to identify the relationship between the songs and the user emotions from the philosophical, psychological, and anthropological perceptions. Shan et al. [6] found that the user's emotion and the user's music listening history are influenced by the time of the day (e.g., morning, afternoon, evening, night, etc.). Bogdanov et al. [7] performed psychological analysis of the users to understand the users' emotions at different times of the day. This psychological analysis is used to learn the user's preference for recommending songs to the user. An alternative method for learning the user's preference is by manually classifying the songs with user emotions. Nevertheless, both psychological analysis and manually classifying the songs with emotions are time-consuming tasks.

Krizhevsky, Sutskever, and Hinton [8] proved that the usage of convolutional neural networks (CNN) gives higher image classification accuracy when compared to other machine learning techniques such as support vector machines (SVM) [9]. This discovery encouraged many researchers to use the CNN for image classification [10–13]. Girshick et al. [10] proposed a regional CNN (RCNN) approach for detecting rich hierarchical features from images. The authors used the RCNN approach for accurately detecting the objects available in the given images [10], whereas Vinyals et at. used the CNN along with the recurrent neural networks (RNNs) to provide image captioning on the dataset used in the 2015 MSCOCO Image Captioning Challenge [11]. The authors used the CNN approach to extract the hidden features of fixed length from the input image [11]. These hidden features are given as an input to the RNN for generating captions for the input image. Hence, Vinyals et al. showed a methodology where the CNN can be used to extract the hidden features from an input image.

Due to the successful usage of the CNN in the field of image classification and image captioning, researchers extended the usage of deep convolutional neural networks (DCNN) for automatic music classification [12,13]. Oord, Dieleman, and Schraumen [12] proposed the usage of the DCNN approach for automatically classifying the songs into different genres. The authors represented the audio signal presented in the song into the log-compressed mel spectrograms. The DCNN uses the mel spectrogram of the song for classifying the songs into different genres. Oord, Dieleman, and Schraumen used the DCNN approach as a scientific way to classify the songs into their genres based on the audio signal presented in the songs. They do not use the metadata (e.g., album name, singer name, popularity, duration, etc.) presented in the songs for classifying the songs. Oord, Dieleman, and Schraumen proved that by using the DCNN approach we can classify the song on different genres based on the audio signals presented in the song. Salamon and Bello [13] proposed usage of the DCNN approach for environmental sound classifications (animals, natural sounds, water sounds, etc.) However, above-mentioned DCNN approaches are limited to classifying the songs into different genres based on the audio signals presented in the songs. They do not extract the latent music features from the audio signal presented in that song. They do not concentrate on how to use the metadata presented in the song for extracting latent music features. Therefore, in this paper, we investigate a personalized music recommendation system (PMRS), based on the DCNN approach to extract latent features from the metadata and the audio signal presented in the song. In the PMRS, we convert the audio signal presented in the song into the mel spectrograms as shown in [12]. We give these mel spectrograms and the metadata of the song as the input to the DCNN for extracting the latent features for that song.

In this paper, we investigate an emotion-aware PMRS (EPMRS) to recommend songs of different genres based on the user's preference and the user's current emotion. The EPMRS uses music website crawlers to crawl the music data from the music websites. The crawled music data contains the metadata and the audio signals of the songs. Figure 1 shows the strategy of the EPMRS. From Figure 1, it can be depicted that the EPMRS provides music recommendations by correlating the user's data and the music data, which are stored in the database. The user's data contains the user's music listening history and the user's information (such as age, gender, location, ambient, time of the data, emotions, etc.). The EPMRS maintains the user's data for each user who is using the system in the database. Based on the two types of data available, the EPMRS uses two approaches: the DCNN approach and the weighted feature extraction (WFE) approach. The DCNN approach classifies the music data based on the metadata and the audio signals presented in the songs. The DCNN approach uses the data presented in the million-song dataset [14] for classifying the songs. We propose a WFE approach for extracting the latent features from the users to music relationships presented in the user data. The EPMRS combines the outputs of the WFE and the DCNN to learn the user's preference to recommend music according to the user's current emotion.

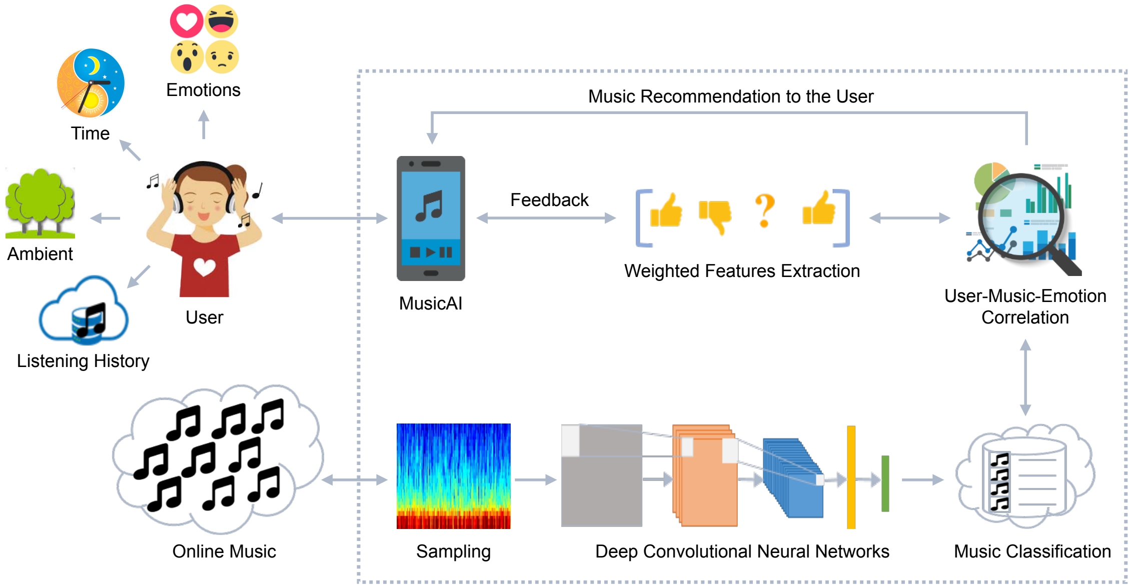

Emotions
Time
Ambient
User
Listening History
Music Recommendation to the User
Feedback
MusicAI
Weighted Features Extraction
User-Music-Emotion Correlation
Online Music
Sampling
Deep Convolutional Neural Networks
Music Classification

Figure 1. The system architecture of the EPMRS (emotion-aware personalized music recommendation system).

The organization of this paper is as follows. In Section 2, we investigate a brief study of related works for the existing PMRS algorithms. We present the data flow design and the system architecture required for the EPMRS as Section 3. However, in Section 4, we show the mathematical model that we use in the EPMRS. In Section 5, we analyze the performance evaluation of the EPMRS. Finally, we conclude this paper in Section 6.

2. Related Works

Researchers used SVM [9] and linear regression [15] to classify the songs based on the audio features of the songs [16,17]. They used traditional approaches such as the mel-frequency cepstral coefficients (MFCCs) for extracting audio features from the songs. Schedl et al. [16] proposed the text mining approach to calculate the artist similarities to classify the songs. Humphrey, Bello, and LeCun [17] proposed that by extracting latent features from the audio signals presented in the songs provides a better classification. Traditional approaches such as MFCCs do not include the metadata (such as artists_familiarity, artist_location, duration, mode, year, tempo, song_id, etc.) presented in the music to classify the songs. The music tracks were classified into positive and negative classes based on the latent features extracted from the songs in [18]. Based on these classifications, they extracted the relationship between the artist and the music track. Social tagging services, such as Last.fm [19] allow users to provide tags describing the genres, moods, instrumentation and locations for classifying the songs. Ignatov et al. [20] proposed a method that correlates the social tags and the keywords mined from the artist profiles to calculate artist relationship scores. These approaches use traditional approaches to extract latent music features to understand users to music relationship. However, these approaches are time-consuming and involves a lot of user interference.

Researchers proved that the using the DCNN for extracting latent music features gives better performance when compared to traditional approaches [12,13]. They proved that the DCNN approaches outperforms the traditional machine learning techniques such as the SVM [9] and the linear regression [15] in terms of classifying songs. The deep neural networks (DNN) approach such as the DCNN [12], the gated recurrent unit (GRU) [21] and the long short-term memory (LSTM) [21] have the capability to work on the huge amount of data in a distributed manner. Oord, Dieleman, and Schraumen [12] proposed usage of the DCNN approach for classifying the songs by identifying the latent music features presented in the songs. Salamon and Bello [13] proposed usage of DCNN for environmental sound classifications. The existing PMRS algorithms [12,13] are limited to recommending songs based on latent music features presented in the user's music listening history. The latent music features for each song are obtained from the audio signal presented in that song. In the proposed EPMRS, we extract the latent features presented in the user's data (containing the user's information and the user's music listening history) and the music data. The EPMRS uses these latent features to recommend songs to the user. In Section 3, we discuss the system architecture and the data flow of the EPMRS.

3. Emotion-Aware PMRS

In this section, we investigate the system architecture and the data flow design of the proposed EPMRS. The EPMRS system architecture is a cloud-based architecture containing three layers: user input layer, request handler layer and the EPMRS layer. All of the three layers presented in the EPMRS system architecture work independently on separate platforms. This independent functionality of the EPMRS system architecture allows us to upgrade or replace any of the three layers independently in response with respect to any technological requirements. Thus, the three-layer EPMRS system architecture provides reliability with well-defined interfaces. Figure 2 shows the three-layered EPMRS system architecture.

- *User input*: This is the front-end layer of the EPMRS system architecture. This layer keeps track of the user's data. The user's data contains the user's music listening data, the user's current emotion, the user's login data, the time of the day, the geographical location, the user's click stream, the user's rating for the song, the ambience, etc. This user's data are given as input data to the request handler layer for further processing. This layer accepts the recommended songs list generated from the EPMRS layer from the request handler layer. The recommended songs' list is displayed to the user as part of the music recommendations.

- *Request handler*: This layer acts as a middle layer between user input and the EPMRS layer. The user's data obtained from the user input layer is the input data to the EPMRS layer for providing music recommendations with respect to the user. This layer gets the recommended songs from the EPMRS layer and sends it to the user input layer.
- *EPMRS*: This layer is the core layer where the EPMRS is deployed. This layer is a cloud-based distributed storage and processing layer. In order to store a huge amount of user behaviors on large scale music (e.g., more than one million songs data), we use the Docker technology to provide cloud environment to the EPMRS. Each Docker container is an Apache spark node containing mongoDB as the database. The Apache spark with mongoDB provides the distributed data storage and processing in this layer. Along with users' data, this layer is also responsible for storing the music data. The music data is collected from the music website crawlers. We use Apache Nutch based music website crawlers to crawl the music data from the online music websites. The music data contains the metadata and the audio signal of the songs. The music data and the user's data are stored as part of the distributed storage in the EPMRS. After processing the user's data and the music data, the EPMRS generates the recommended songs list to the user. This recommended songs list is given as input to the request handler layer.

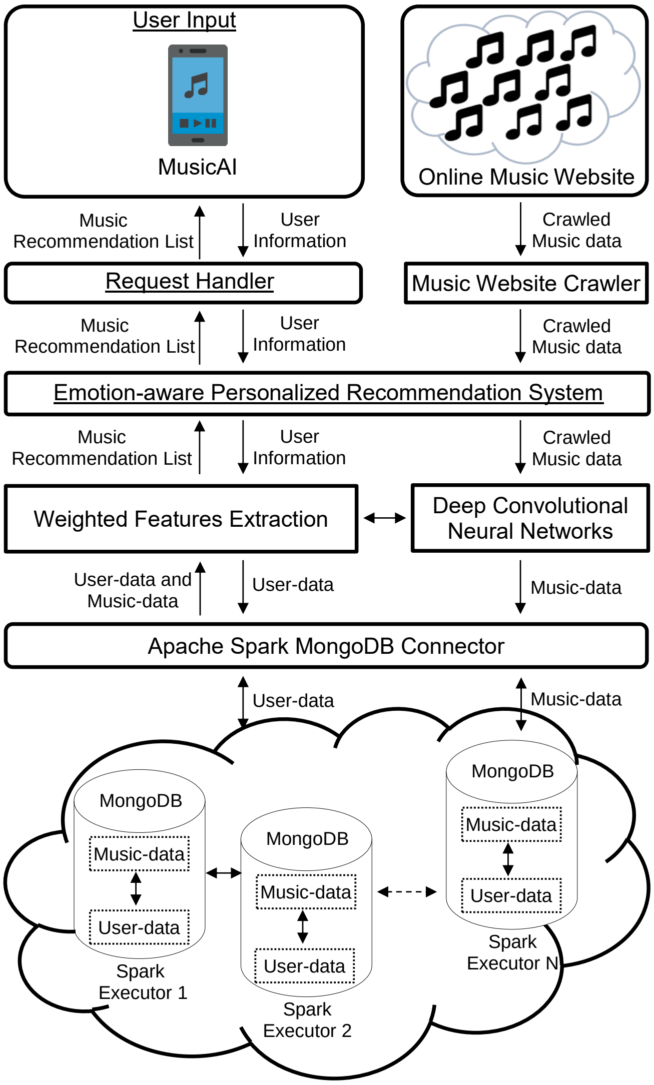


Figure 2. The system architecture and the data flow for the EPMRS.

According to the two types of data received at the EPMRS layer, the EPMRS uses two approaches: the WFE approach and the DCNN approach. We propose a WFE approach to extract the user's latent features presented in the user's data. The user's latent features extracted from the user's data contain the details about the user-to-song relationship with respect to the user's current emotion. The user's latent features are stored as part of the user data in the database. The DCNN approach classifies songs available in the crawled music data based on the metadata and the audio signal presented in the songs. The DCNN approach stores the songs according to their classification as part of the music data in the database. In Section 4, we discuss the mathematical model for the EPMRS.

4. EPMRS Mathematical Model

In this section, we describe the mathematical model for the EPMRS with the help of the WFE approach and the modified DCNN approach. In Section 4.1, we discuss about the million song dataset (MSD) [14] used for developing the EPMRS. The mathematical model for WFE approach is discussed in Section 4.2 and the modified DCNN approach is described in Section 4.3.

4.1. Dataset

In EPMRS, we use the MSD [14] to train the system for songs' recommendations. The MSD contains precomputed audio features and metadata of one million songs. This dataset is of size 250 GB containing the details of songs sung by 44,745 unique artists with 2321 tags dated since 1922. Apart from this, the MSD also provides the similarity relationships among the artists. Each song is described as a set of 55 fields, which includes artists_familiarity, artist_location, audio_md5, duration, mode, year, tempo, song_id, time_signature, title, track_id, artist_name, beats_starts, track_digitalid, etc. Due to the size and the variety of data presented in the MSD, researchers of MIR use it for validating their song recommendations' algorithms. This dataset is deployed on the system architecture discussed in Section 3.

4.2. Weighted Feature Extraction

The WFE approach uses the weighted matrix factorization technique [22]. The WFE approach extracts the latent features from user-to-song relationships presented in the user's data. A record in the user's data contains these elements: <UID, SID, ET, TD, LOC, PC, ER, IR, OI>, where the UID is the user identification, the SID is the song identification, ET is the user emotion type, the TD is time of the day when the user listens to the song, LOC is the user's current location, the PC is the play count of the song, the ER is the explicit rating, the IR is the implicit rating, and the OI is the overall interest with respect to a user for the song, respectively. Notice that the ET element we consider in the EPMRS has four user emotion types: happy, normal, sad, and surprised. The types of TD we consider in the system are morning, noon, afternoon, evening, and midnight. The classes of LOC we consider are home, work place, and others. Let $p_t(u,s)$ denote the number of times the user u has listened to the song s at time t. If the user u listens to the song s, we have $p_{t+1}(u,s) = p_t(u,s) + 1$.

For getting the feedback from user to the songs, we consider two rating mechanisms: the explicit rating mechanism and the implicit rating mechanism. In the explicit rating mechanism, a user rates the song after listening to the song in the form of like and dislike. Let $e_t(u,s) = \{0,1,2,3,4,5\}$ denote the explicit rating (i.e., dislike (0) or like (1 to 5 stars) given by a user u to a song s at time t. The $e_t(u,s)$ represents the ER element of the user's data record. If the user does not provide any explicit rating to a song after listening to a song, then the EPMRS will provide an implicit rating to the song. Let $i_t(u,s)$ denote the implicit rating given by the EPMRS to a song on behalf of user u at time t. If the user u listens to more than half of the duration of s, then the EPMRS considers that the user likes the song. We have $i_{t+1}(u,s) = i_t(u,s) + 1$ if the listening time is more than the half of the duration of s.

At the beginning of learning a user's preference on music, the values of $e_t(u,s)$ and $i_t(u,s)$ may be equal to zero. It may lead to two confusing situations: either the user u is not aware of the song s or the user u is not interested in listening to the song s. To remove this confusion, we introduce an overall user's interest variable. The overall user's interest variable $o_t(u, s)$ calculates how much the user u will be interested to listen the song s at time t. The value of $o_t(u, s)$ can be obtained by

$$o_t(u, s) = \frac{p_t(u, s) + i_t(u, s)}{2} + e_t(u, s). \tag{1}$$

The EPMRS stores the user's data as a part of the user data in the database.

Let $U = \{u_1, u_2, \ldots, u_n\}$ denote the set of n users and $|U| = n$ is the size of U. Let $S = \{s_1, s_2, \ldots, s_m\}$ denote the set of m songs and $F = \{f_1, f_2, \ldots, f_v\}$ denote the set of v features of each song in S. Let $N = (n_{i,j}) \in \mathbf{Z^+}^{n \times m}$ denote the user-to-song interest matrix of size $|U| \times |S|$. The value of each element of matrix N, $(n_{i,j})$, is the value of $o_t(u, s)$. In order to better understand the working process of the EPMRS, we take an example of six users and five songs to illustrate our method. In the example, the interest matrix N is shown as

$$N = \begin{array}{c} \\ u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \end{array} \begin{array}{c} \begin{array}{ccccc} s_1 & s_2 & s_3 & s_4 & s_5 \end{array} \\ \begin{pmatrix} 3 & - & - & 5 & - \\ 2 & - & 5 & 4 & - \\ 4 & - & 2 & 2 & 5 \\ - & 3 & - & 1 & - \\ 4 & 1 & - & 5 & 3 \\ 2 & - & 4 & - & 5 \end{pmatrix} \end{array},$$

where the '-' symbol indicates a null value. Please notice that all values shown in the matrix are made up by the authors for illustration. In practice, each value in N is obtained by Equation (1). Please notice that the null value implies that a user u_i has not listened to a song s_j or the song s_j is a new song. We note that the EPMRS has different matrices N based on how many types of ET, TD, and LOC the system has. The number of N is equal to |ET| × |TD| × |LOC|, i.e., |ET| = 4, |TD| = 5, and |LOC| = 3 in the system because the EPMRS will record the users' behaviors under different emotions, the time of the day, and the location's conditions. For instance, N(happy, evening, home) representing one matrix means the users' preferences on songs under the condition of being happy, in the evening, and at home. We note that the selection of N with the conditions of being happy, evening, and home is based on the current data stored in the database. Thus, the result may be a null matrix.

Let matrix $M = (m_{j,k}) \in \{0, 1\}^{n \times v}$ denote a binary matrix that the songs correspond to their latent features. In the example, we assume each song has four latent features and we have

$$M = \begin{array}{c} \\ s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \end{array} \begin{array}{c} \begin{array}{cccc} f_1 & f_2 & f_3 & f_4 \end{array} \\ \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 \end{pmatrix} \end{array}.$$

For instance, $M_{1,j} = [1\ 0\ 1\ 0]$ indicates that s_1 has two features f_1 and f_3 and does not have features f_2 and f_4.

We use matrices N and M to generate the matrix $Q = (q_{i,k}) \in \mathbf{Z^+}^{n \times v}$ representing the user-to-feature relationship of size $n \times v$ as

$$Q = (q_{i,k})^{n \times v} = \sum_{j=1}^{|S|} m_{j,k}, \forall n_{i,j} > T_s, \tag{2}$$

where $i = 1, 2, \ldots, n$, $j = 1, 2, \ldots, m$, $k = 1, 2, \ldots, v$, and T_s is the threshold value for overall interest for the song s. In EPMRS, we set $T_s = 1$ as the default value:

$$Q = \begin{array}{c} \\ u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \end{array} \begin{array}{c} \begin{array}{cccc} f_1 & f_2 & f_3 & f_4 \end{array} \\ \begin{pmatrix} 1 & 0 & 2 & 0 \\ 2 & 2 & 3 & 1 \\ 2 & 1 & 4 & 1 \\ 1 & 2 & 0 & 2 \\ 3 & 1 & 3 & 2 \\ 2 & 1 & 3 & 1 \end{pmatrix} \end{array}.$$

The value $q_{1,1} = 1$ denotes that the feature f_1 has appeared one time in the songs listened by the user u_1, whereas the value $q_{1,2} = 0$ indicates that the feature f_2 is not present in any of the songs listened to by the user u_1. The matrix Q contains the relationship between each user u_i and all features, which includes some dominant features and some unimportant features with respect to the user u_i. Hence, we apply weights on the features presented in the matrix Q to extract these dominant features with respect to each user u_i. We use the term frequency and inverse document frequency (TF-IDF) [23] approach as the user feature frequency $F_U(f_k)$ and the inverse user frequency $F_{U^{-1}}(f_k)$ to generate the important features from the matrix Q. The $F_U(f_k)$ represents which feature f_k is most relevant to the user u_i and is calculated as

$$F_U(f_k) = \sum_{i=1}^{|U|} 1, \forall q_{i,k} > 0, k = 1, 2, \ldots, v. \tag{3}$$

Let $F_U^{-1}(f_k)$ represent how important a feature f_k is with respect to u_i and this is calculated as

$$F_U^{-1}(f_k) = \log\left(\frac{|U|}{F_U(f_k)}\right). \tag{4}$$

We note that $F_U(f_k) = F_U(f_k) - 0.1$ if $F_U(f_k) = |U|$. Let $W = (w_{i,k}) \in \mathbf{Z^+}^{n \times v}$ denote the matrix of the weighted representation of Q and this is obtained by

$$W = (w_{i,k})^{n \times v} = F_U(f_k) \cdot F_U^{-1}(f_k), \forall u_i \in U, \tag{5}$$

where $i = 1, 2, \ldots, n$ and $k = 1, 2, \ldots, v$. Following the example, we apply Equation (5) on the matrix Q to generate $W = (w_{i,k}) \in \mathbf{Z^+}^{6 \times 4}$ as

$$W = \begin{array}{c} \\ u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \end{array} \begin{array}{c} \begin{array}{cccc} f_1 & f_2 & f_3 & f_4 \end{array} \\ \begin{pmatrix} 0.007 & 0 & 0.158 & 0 \\ 0.014 & 0.158 & 0.238 & 0.079 \\ 0.014 & 0.079 & 0.318 & 0.079 \\ 0.007 & 0.158 & 0 & 0.158 \\ 0.021 & 0.079 & 0.238 & 0.158 \\ 0.014 & 0.079 & 0.238 & 0.079 \end{pmatrix} \end{array}.$$

Based on Equation (6), we can easily determine the dominant feature of u_i by selecting the maximal value of $w_{i,k}$ where $k = 1, 2, \ldots, v$. For instance, the value $w_{2,3} = 0.238$ represents that the feature f_3 is the dominant feature of the user u_2.

Let matrix $\overline{M} = (\overline{m}_{j,k}) \in \mathbf{Z^+}^{m \times v}$ be a new song-to-feature matrix containing the songs that are not listened by $u_i \in U$, where $\overline{M} \subset M$. Following the example of matrices N and M, let us consider that user u_1 has not listened to the songs presented in the matrix $\overline{M} = (\overline{m}_{j,k}) \in \mathbf{Z}^{+2\times4}$ with two songs and four features as

$$\overline{M} = \begin{array}{c} \\ s_5 \\ s_8 \end{array}\begin{array}{c} \begin{array}{cccc} f_1 & f_2 & f_3 & f_4 \end{array} \\ \begin{pmatrix} 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{pmatrix} \end{array}.$$

Now, we calculate the $o_t(u,s)$ values for $u_i \in U$ with respect to the songs presented in matrix $\overline{M}$ and generate a weighted user-to-song matrix $\overline{N} = (\overline{n}_{i,j}) \in \mathbf{Z}^{+n\times m}$ as

$$\overline{N} = W\overline{M}^T. \tag{6}$$

Based on $\overline{N}$, the EPMRS can easily recommend the songs to u_i by selecting the maximum value of $\overline{n}_{i,j}$ where $j = 1, 2, \ldots, m$. Following the above-mentioned example matrix W with respect to the user u_1, we apply $\overline{M}$ on Equation (6) to obtain $\overline{N} = (\overline{n}_{i,j}) \in \mathbf{Z}^{+1\times2}$ as

$$\overline{N} = u_1 \begin{array}{c} \begin{array}{cc} s_7 & s_8 \end{array} \\ \begin{pmatrix} 0.238 & 0.079 \end{pmatrix} \end{array}. \tag{7}$$

Finally, the EPMRS will recommend the song s_5 to u_1 as the value $\overline{n}_{1,1} = 0.238$ is the maximum value in the record of $\overline{N}$ of Equation (7).

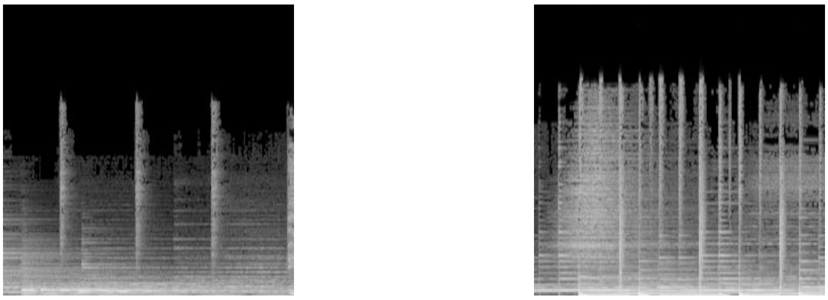
Let $l_u = \{l_{u_1}, l_{u_2}, \ldots, l_{u_n}\}$ denote the set of the latent features of $u_i \in U$ and $l_s = \{l_{s_1}, l_{s_2}, \ldots, l_{s_m}\}$ represent the set of the latent features of $s_j \in S$. The optimized WFE function is

$$\min_{u_i, s_j} \sum_{u_i, s_j} o_t(u_i, s_j)\left(\overline{N} - l_{u_i}^T l_{s_j}\right)^2 + \lambda \left(\sum_{u_i} ||l_{u_i}||^2 + \sum_{s_j} ||l_{s_j}||^2\right), \tag{8}$$

where λ is the regularization parameter, $i = 1, 2, \ldots, n$, and $j = 1, 2, \ldots, m$. The optimized WFE function as shown in the Equation (8) consists of two terms: the weighted mean square error term and the $L2$ normalization term. The first term of the Equation (4) calculates the weighted mean square error of all the users and all songs in the database. The overall interest variable $o_{u,s}$ gives weights to all possible combinations that may also contain songs that are not rated by the user u. The second term of the Equation (8) normalizes the latent features extracted from the user u and the song s.

4.3. Deep CNN

The DCNN approach classifies the songs based on the metadata and the audio signals presented in the songs. The first step in classifying the crawled music data is to identify the necessary components of the audio signal (such as the linguistic content) and discarding the unnecessary content (such as noise etc.). In order to classify the crawled music data, we first extract the MFCCs [12] presented in the audio signal of the song. In order to extract the MFCCs, we sample the audio signal presented in the song into the audio clips of 2 s. To speed up the training time of the network, we can increase the size of the sampled audio clips. We use the log-powered mel-spectrograms with 128 components to extract the MFCCs from the sampled audio clips of the song. Figure 3a shows the mel-spectrograms of the starting 2 s sampled audio clip for a song with the Downtempo genre. In Figure 3b, we display the mel-spectrograms of the starting 2 s sampled audio clip for a song with the Rock genre.



(**a**) (**b**)

Figure 3. (**a**) the mel-spectrogram of an audio signal for a song with the *Downtempo* genre; (**b**) the mel-spectrogram of an audio signal for a song with the *Rock* genre.

Figure 3 shows the difference between the audio signals obtained for the songs with different genres. Apart from the audio signal of the songs, the metadata (e.g., album name, singer name, popularity, duration, etc.) associated with the songs are also different. Therefore, in the EPMRS, the DCNN approach extracts the latent features for a song from the metadata of the song and the MFCCs obtained for that song. The DCNN approach uses these latent music features from the music data to classify the songs. The latent music features for entire audio signal is the average of the latent features extracted from each audio clip for that audio signal. In the EPMRS, we increase the performance of the DCNN as follows:

- Rectified linear units (RLUs) are used as an alternative to the sigmoid function. The usage of RLUs in DCNN contributes to faster convergence. The RLUs minimizes the vanishing gradient problem, which is a common problem in traditional multi layer neural networks.
- In order to increase the speed of the EPMRS, we execute the DCNN approach in a parallel fashion on the GPU. We used the Keras library [24] to execute the DCNN on the GPU in a parallel fashion.

Let $\hat{l_s} = \{\hat{l_{s_1}}, \hat{l_{s_2}}, \ldots, \hat{l_{s_m}}\}$ denote the set of the latent features extracted for the songs $s_j \in S$ using the DCNN approach. The objective function of the EPMRS is to minimize the mean square error (MSE) for predicting the latent features of the users and the songs. Based on the WFE and the DCNN approaches, we can continuously minimize the MSE as

$$\min_{s_j} \sum_{s_j} ||l_{s_j} - \hat{l_{s_j}}||^2, \tag{9}$$

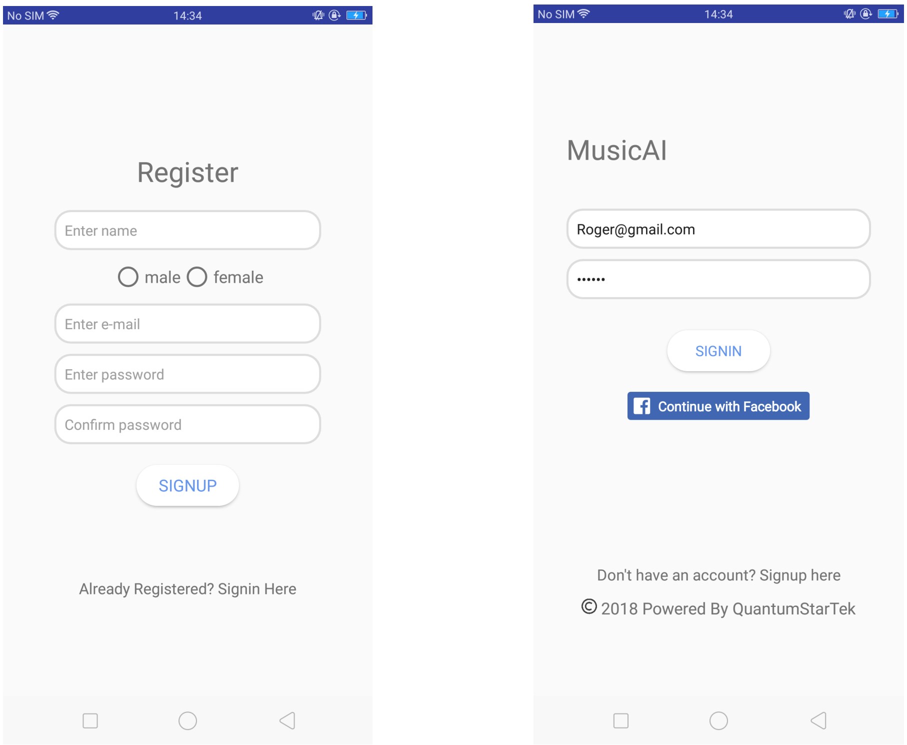
and

$$\min_{u_i,s_j} \sum_{u_i,s_j} \left(\overline{N} - l_{u_i}^T \hat{l_{s_j}}\right)^2, \tag{10}$$

where $i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, m$. These optimized latent features of the users and the songs are stored in the database. In the EPMRS, these latent features are updated continuously based on the users' music listening history and the users' information for providing better music recommendations to the users.

5. Experimental Results

For demonstrating the function of the EPMRS, we developed an Android-based application named MusicAI. MusicAI is built by using the Python programming language. The DCNN approach is developed by the Keras module of the Python programming language. MusicAI is an end-user APP installed in the smart phone device for providing users' online music listening. First, to trace the individual user listen behavior, we request all users to register an account once they open MusicAI at the first time. There are two ways to register a valid account in MusicAI: through the formal registration approach or through the user's Facebook account. Figure 4a shows a screenshot of the user's registration for MusicAI. These registration details are stored as a part of the user-data in the database. A user should provide the user credentials to login into the MusicAI application as shown in Figure 4b.



Register

Enter name

○ male ○ female

Enter e-mail

Enter password

Confirm password

SIGNUP

Already Registered? Signin Here

MusicAI

Roger@gmail.com

••••••

SIGNIN

Continue with Facebook

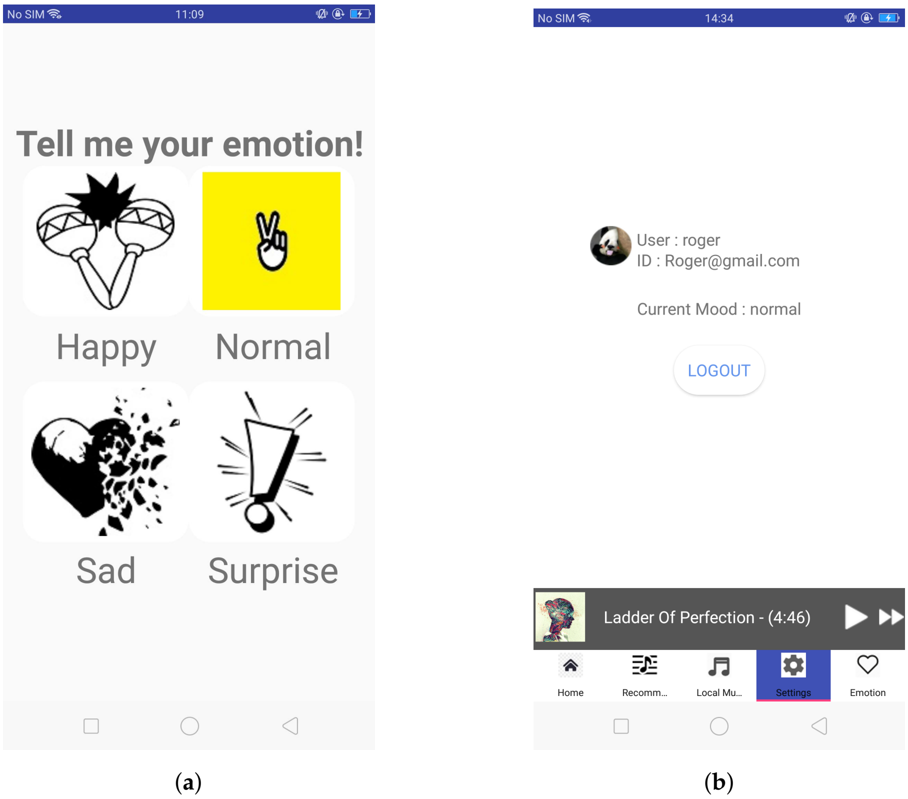
Don't have an account? Signup here

© 2018 Powered By QuantumStarTek

(**a**) (**b**)

Figure 4. (**a**) registration form a user to get the user details; (**b**) user's login page.

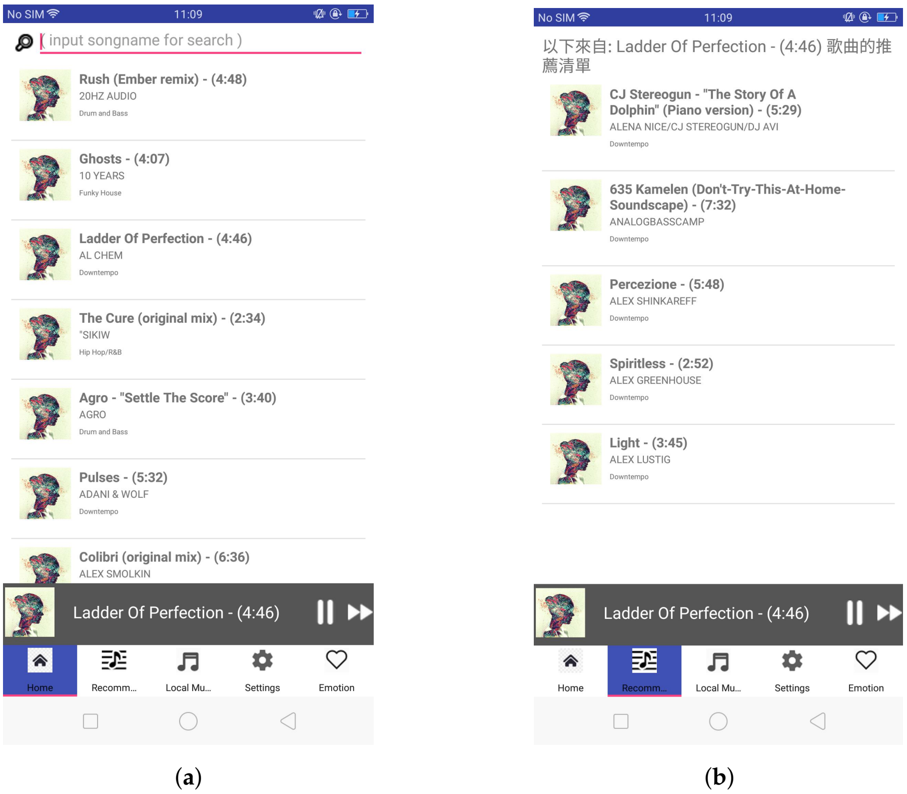
After successful user login, the MusicAI asks the user to provide his current emotion as shown in Figure 5a. In MusicAI, we consider four user emotions: happy, normal, sad, and surprise. It is compulsory for the user to select one of the four user emotions before using any functionality of the MusicAI. By adding this constraint, we assume that a user will provide a *true* user's emotional state. Once the user selects his current emotion, the MusicAI displays a list of songs to the user for listening as shown in Figure 6a. We emphasize that users can change their current emotions at any time later if they want to change the recommended songs based on their emotions. The list of songs is a collection of songs that is stored on user's device and the database of the MusicAI. The MusicAI application monitors the user behavior such as his current listening history, the genre of the selected song, the user's click stream, etc. The user information including his current emotion can be seen and shown in Figure 5b. The MusicAI application sends the user's data to the EPMRS for analysis. The EPMRS analyzes the user's data to generate a list of recommended songs. The MusicAI application displays the list of recommended songs received from the EPMRS to the user as shown in Figure 6b. We use the MusicAI to trace the user's information (e.g., the emotion, the location, the time of play, etc.) and monitor the listen history of the user. The MusicAI sends all data collected from the user to the EPMRS for deep learning. Based on the built DCNN model and the collected data from users, the EPMRS recommends songs to the user.



User : roger
ID : Roger@gmail.com

Current Mood : normal

Figure 5. (**a**) list of emotions where a user can select his current emotion; (**b**) user information.



Rush (Ember remix) - (4:48)
20HZ AUDIO
Drum and Bass

Ghosts - (4:07)
10 YEARS
Funky House

Ladder Of Perfection - (4:46)
AL CHEM
Downtempo

The Cure (original mix) - (2:34)
"SIKIW
Hip Hop/R&B

Agro - "Settle The Score" - (3:40)
AGRO
Drum and Bass

Pulses - (5:32)
ADANI & WOLF
Downtempo

Colibri (original mix) - (6:36)
ALEX SMOLKIN

以下來自: Ladder Of Perfection - (4:46) 歌曲的推薦清單

CJ Stereogun - "The Story Of A Dolphin" (Piano version) - (5:29)
ALENA NICE/CJ STEREOGUN/DJ AVI
Downtempo

635 Kamelen (Don't-Try-This-At-Home-Soundscape) - (7:32)
ANALOGBASSCAMP
Downtempo

Percezione - (5:48)
ALEX SHINKAREFF
Downtempo

Spiritless - (2:52)
ALEX GREENHOUSE
Downtempo

Light - (3:45)
ALEX LUSTIG
Downtempo

Figure 6. (**a**) music list available on the user's device and the MusicAI's server; (**b**) music recommendations' list based on the user's information and his current listening history of songs.

In EMPRS, we measure the performance in terms of the accuracy of the songs' recommendation with respect to different genres. Therefore, we propose the accuracy formula, $A_{u,s}$, of recommending a song s to a user u as

$$A_{u,s} = \frac{\sum \text{the number of correct song recommendations}}{\sum \text{the total number of song recommendations}} \times 100. \tag{11}$$

In order to compare the performance of the EPMRS, we adopt two baseline systems: the content similarity for music recommendation system (CSMRS) [25] and the personalized music recommendation system based on electroencephalography feedback (PMRSE) [26]. The CSMRS uses a CB recommendation approach to extract the latent features from the audio content of the song. The CSMRS extracts the MFCCs from the audio signal and aggregates them into a bag-of-words representation. The PMRSE uses the K-MeansH clustering algorithm for clustering the music based on the content presented in the music. Apart from the K-MeansH clustering algorithm, the PMRSE also uses the CF based recommendation system for providing personalized music to recommend to the user. In order to implement the CSMRS, we downloaded the python code available at the github repository [27], whereas, in case of the PMRSE, we modified the python code available at the GitHub repository [28] with respect to the K-MeansH clustering algorithm.

For training the EPMRS, we use the data presented in the MSD (Section 4.1). As the MSD is a huge dataset, we selected the songs of nine genres: breakbeat, dancehall, downtempo, drum and bass, funky house, hip hop, minimal house, rock, and trance. From each genre, we selected the top 3000 songs to train the DCNN approach. To train the WFE approach, we used the user data available in the 3000 songs of each genre. To test the music genre classification, we crawled 1000 songs for each of nine genres from the JunoRecords website [29]. We call this dataset the Junorecords dataset. The Junorecords dataset contains the music data but does not contain user data. We randomly selected 100 users data (e.g., user rating on songs, user listening behavior, etc.) from the MSD and associated that users data with music data (e.g., audio signals of songs, the metadata of songs, etc.) of the Junorecords dataset. Once the user data are selected, all methods—the EPMRS, the CSMRS, and the PMRSE—use the selected user data for performance comparison.

Table 1 shows the mean accuracy of song recommendations in the nine genres for the 100 users. The experimental results show that the EPMRS gets significant improvement in the song recommendation accuracy except the genres: the funky house and the minimal house as compared with the CSMRS. Although the EPMRS does not outperform the CSMRS in funky house and minimal house genres, the EPMRS still obtains overall improvement than the CSMRS. The performance of the PMRSE approach is lower as compared with the EPMRS and the CSMRS approaches for the nine genres. Unlike the CSMRS and the PMRSE, the EPMRS uses the audio content and the metadata presented in the songs' data to extract latent features for 100 users. Even though the PMRSE approach uses the CF recommendation system, the performance of the EPMRS is better as it uses the WFE approach. The WFE approach of the EPMRS identifies the important latent features from the data and provides weights to those features. Therefore, for each user, the EPMRS maintains a separate weighted users-to-features relationship data. The appropriate songs containing the user's weighted features are recommended to the user. The CSMRS and the PMRSE do not have any mechanism for extracting weighted features with respect to each user. This is also the main reason why the EPMRS shows better accuracy of song prediction than the CSMRS and the PMRSE. In the EPMRS, the user's latent features are updated continuously from the user's music listening behavior. Therefore, the MusicAI can provide better recommendations to the users by continuously learning their music listening behavior.

Table 1. The accuracy (%) of song recommendations in the nine genres for the 100 users.

Genre	EPMRS	CSMRS	PMRSE
Breakbeat	74.96	60.66	52.33
Dancehall	94.56	80.44	73.15
Downtempo	76.52	61.91	50.98
Drum and bass	85.56	80.43	75.32
Funky house	79.55	82.45	74.53
Hip Hop	84.56	81.12	78.84
Minimal house	69.32	73.56	62.37
Rock	95.36	90.56	68.04
Trance	85.28	75.89	65.74

6. Conclusions

In this paper, we have shown how to implement a personalized song recommendation system based on the user′s time, ambience, preference, geographical location, user′s current emotion, user′s song listening behavior, play count of the songs, duration of the audio track, etc. We ask the user to provide his current emotion as the input to the EPMRS. The experiment results show that an individual user′s preference for songs with respect to user′s current emotions can be learned by using the DCNN and the WFE approaches. The EPMRS uses the user-to-song relationship to recommend songs to the user based on the user′s current emotion. As part of the future work, we would like to use the user′s social media data to extract the user′s current emotion automatically. For better understanding of the user′s preference, we will consider the user′s data from other sources such as YouTube, Facebook, Twitter, and so forth. Another potential way of improving the performance of the EPMRS is to consider using recurrent neural network instead of DCNN for song classification.

Author Contributions: J.C. proposed the research direction and gave the conceptualization. A.A. implemented the DCNN and the WFE approach. S.-H.C., and H.-Y.L. implemented the application. A.A., J.C., S.-H.C., and H.-Y.L. performed the verification and analyzed the results. J.C. and A.A. wrote the paper.

Funding: This work was supported in part by the Ministry of Science and Technology, Taiwan, under Contract MOST 106-2221-E-182-065.

Conflicts of Interest: The authors declare that there is no conflict of interests regarding the publication of this article.

Abbreviations

The following abbreviations are used in this manuscript:

CB	content based
CF	collaborative filtering
CSMRS	content similarity music recommendation sys
DCNN	deep convolutional neural networks
EPMRS	emotion-aware personalized music recommendation system
GRU	gated recurrent units
LSTM	long short term memory
PMRS	personalized music recommendation system
PMRSE	personalized music recommendation system based on electroencephalography feedback
RNN	recurrent neural networks
SVM	support vector machine
WFE	weighted feature extraction

References

1. Hyung, Z.; Park, J.S.; Lee, K. Utilizing context-relevant keywords extracted from a large collection of user-generated documents for music discovery. *Inf. Process. Manag.* **2017**, *53*, 1185–1200, . [CrossRef]
2. Pazzani, M.J.; Billsus, D. Content based recommendation systems. In *The Adaptive Web: Methods and Strategies of Web Personalization*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2007; pp. 325–341.
3. Schafer, J.B.; Frankowski, D.; Herlocker, J.; Sen, S. Collaborative filtering recommender systems. In *The Adaptive Web: Methods and Strategies of Web Personalization*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2007; pp. 291–324.
4. Burke, R. Hybrid web recommender systems. In *The Adaptive Web: Methods and Strategies of Web Personalization*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2007; pp. 377–408.
5. Aljanaki, A.; Wiering, F.; Veltkamp, R.C. Studying emotion induced by music through a crowdsourcing game. *Inf. Process. Manag.* **2016**, *52*, 115–128 . [CrossRef]
6. Shan, M.K.; Kuo, F.F.; Chiang, M.F.; Lee, S.Y. Emotion-based music recommendation by affinity discovery from film music. *Expert Syst. Appl.* **2009**, *36*, 7666–7674 . [CrossRef]
7. Bogdanov, D.; Wack, N.; Gómez, E.; Gulati, S.; Herrera, P.; Mayor, O.; Roma, G.; Salamon, J.; Zapata, J.; Serra, X. ESSENTIA: An open-source library for sound and music analysis. In Proceedings of the 21st ACM International Conference on Multimedia, Barcelona, Spain, 21 Octobor 2013; pp. 855–858.
8. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In Proceedings of the Advances Neural Information Processing Systems, Lake Tahoe, NV, USA, 3–6 December 2012; pp. 1097–1105.
9. Hsu, C.W.; Lin, C.J. A comparison of methods for multiclass support vector machines. *IEEE Trans. Neural Netw.* **2002**, *13*, 415–425. [PubMed]
10. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
11. Vinyals, O.; Toshev, A.; Bengio, S.; Erhan, D. Show and tell: Lessons learned from the 2015 mscoco image captioning challenge. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 652–663. [CrossRef] [PubMed]
12. Oord, A.V.; Dieleman, S.; Schrauwen, B. Deep content-based music recommendation. In Proceedings of the 26th Neural Information Processing Systems, Lake Tahoe, CA, USA, 5–10 December 2013; pp. 2643–2651.
13. Salamon, J.; Bello, J.P. Deep convolutional neural networks and data augmentation for environmental sound classification. *IEEE Signal Process. Lett.* **2017**, *24*, 279–283. [CrossRef]
14. Bertin-Mahieux, T.; Ellis, D.P.; Whitman, B.; Lamere, P. The million song dataset. *ISMIR* **2011**, *2*, 10.
15. Fan, R.E.; Chang, K.W.; Hsieh, C.J.; Wang, X.R.; Lin, C.J. LIBLINEAR: A library for large linear classification. *J. Mach. Learn. Res.* **2008**, *9*, 1871–1874.
16. Schedl, M.; Pohle, T.; Knees, P.; Widmer, G. Exploring the music similarity space on the web. *ACM Trans. Inf. Syst.* **2011**, *29*, 1–24. [CrossRef]
17. Humphrey, E.J.; Bello, J.P.; LeCun, Y. Moving beyond feature design: Deep architectures and automatic feature learning in music informatics. In Proceedings of the 13th International Society for Music Information Retrieval Conference, Porto, Portugal, 8 October 2012; pp. 403–408.
18. McFee, B.; Lanckriet, G.R. Learning multi-modal similarity. *J. Mach. Learn. Res.* **2011**, *12*, 491–523.
19. Haupt, J. Last.fm: People-powered online radio. *Music Ref. Serv. Q.* **2009**, *12*, 23–24. [CrossRef]
20. Ignatov, D.I.; Nikolenko, S.I.; Abaev, T.; Poelmans, J. Online recommender system for radio station hosting based on information fusion and adaptive tag-aware profiling. *Expert Syst. Appl.* **2016**, *55*, 546–558. [CrossRef]
21. Tan, Y.K.; Xu, X.; Liu, Y. Improved recurrent neural networks for session-based recommendations. In Proceedings of the 1st Workshop on Deep Learning for Recommender Systems, Boston, MA, USA, 15 September 2016; pp. 17–22.
22. Hu, Y.; Koren, Y.; Volinsky, C. Collaborative filtering for implicit feedback datasets. In Proceedings of the 8th IEEE International Conference Data Mining, Pisa, Italy, 15–19 December 2008; pp. 263–272.
23. Wu, H.C.; Luk, R.W.P.; Wong, K.F.; Kwok, K.L. Interpreting TF-IDF term weights as making relevance decisions. *Proc. ACM Trans. Inf. Syst.* **2008**, *26*, 13:1–13:37. [CrossRef]

24. Erickson, B.J.; Korfiatis, P.; Akkus, Z.; Kline, T.; Philbrick, K. Toolkits and libraries for deep learning. *J. Digit. Imag.* **2017**, *30*, 400–405. [CrossRef] [PubMed]
25. McFee, B.; Barrington, L.; Lanckriet, G. Learning content similarity for music recommendation. *IEEE Trans. Audio Speech Lang. Process.* **2012**, *20*, 2207–2218. [CrossRef]
26. Chang, H.Y.; Huang, S.C.; Wu, J.H. A personalized music recommendation system based on electroencephalography feedback. *Multimed. Tools Appl.* **2017**, *76*, 19523–19542. [CrossRef]
27. Jaganmohan. MusicRecommendation: Content Based Recommendation System. GitHub. Available online: https://github.com/jaganmohan/MusicRecommendation/ (accessed on 1 April 2018).
28. Anandbhoraskar. Music Classification. GitHub. Available online: https://github.com/anandbhoraskar/musicClassification/ (accessed on 16 June 2018).
29. Junorecords. Available online: https://www.juno.co.uk/ (accessed on 1 November 2017).