ADDIS ABABA UNIVERSITY

COLLEGE OF NATURAL SCIENCES

MOOD BASED HYBRID ETHIOPIC MUSIC RECOMMENDER

Biruk Gebru Wolde

A Thesis Submitted to the Department of Computer Science in Partial Fulfillment for the Degree of Master of Science in Computer Science

Addis Ababa, Ethiopia

September, 2018

Addis Ababa University

College of Natural Sciences

Biruk Gebru

Advisor: Fekade Getahun (PhD)

This is to certify that the thesis prepared by Biruk Gebru, titled: *Mood Based Hybrid Ethiopic Music Recommender* and submitted in partial fulfillment of the requirements for the Degree of Master of Science in Computer Science complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the Examining Committee:

	Name	Signature	Date
Advisor:	Fekade Getahun (PhD)		
Examiner:			
Examiner:			

Abstract

Music is one of the most engaging and enormously spreading content on the Internet that plays an important role in our daily life. This has created new demand for easier services that support music navigation and discovery. Several music recommenders are proposed to contribute for this demand. However, many research questions are still open. Some mood based music recommenders are proposed but there is no any system considering Ethiopic Music. They favor popular songs that lack awareness of user's contextual situation. They require a lot of user's effort.

Here, we proposed mood based context aware music recommender for smartphone. That has three main tasks, including: 1) Constructing mood based Ethiopic song classifier based on a model trained using linear SVM. 2) User modeling that includes user mood detection module built by combining biometric (heart-rate) and text mood expression modalities using Dempster Shafer theory. 3) Creating an association between user contextual interest and songs to draw list of recommendations. High Positive Affect, Low Positive Affect, Pleasantness, Strong Engagement, and Unpleasantness are the primary moods considered in this study. These has shown accuracy of 65% in song classification, accuracy of 95% in user mood detection and a good feedback gained from subjects that participated in overall evaluation of the recommender. We used 600 Ethiopic Songs and 25,800 mood sentences. Generally the study revealed algorithm and audio features to detect mood of Ethiopic song as well as a new way of user modeling for recommender systems. These can be applied on music information retrieval, music streaming websites, media players and systems that involve user mood detection.

Keywords: Music Mood, Recommender, Dempster Shafer Theory, Mood Detection, Information Retrieval, Soft Clustering, linear SVM

Dedication

For Muluemebet Aman and YeGebruye Lijoch

Acknowledgments

When you focus on chasing your dream, God makes sure your dream comes true.
To God be the Glory.
I would like to thank my advisor Fekade Getahun (PhD), for his welcoming gesture whenever I had a question about this research. He consistently provided valuable guidance, encouragement and advice from the start.

This thesis is a milestone of the effort by the instructors and technicians of Computer Science Department at Addis Ababa University (AAU). It is the result of many experiences I encountered throughout my years of study at AAU. The thesis would not have come to a successful completion, without the help I got from them. They deserve thanks.

A lot of people have been participated in the survey for this research, I would also like to thank everyone who were involved in this survey. Without their passionate participation, the survey could not have been successfully conducted.

I would also like to thank Selam Aregaw, director of Yared School of Music at AAU. She had done her best at pointing important resources and providing information.

Librarians of Yared School of Music at AAU was kind enough to help me whenever I need their help. I really appreciate their interest and effort to add their own contribution for this research, Thank You.

Finally, I would like to acknowledge Dr. Million Dechassa of St. Paul's Millennium Medical College and PhD students of School of Psychology 2015 entry at AAU for their support in one or another at user modeling stage of this research.

Table of Contents

Abbreviations and Acronyms

BBA	Basic Belief Assignment
CAF	Context Aware Filtering
CBF	Content Based Filtering
CF	Collaborative Filtering
DS	Dempster-Shafer
DisE	Disengagement
HCDF	Harmonic Change Detection Function
HNA	High Negative Affect
HPA	High Positive Affect
LNA	Low Negative Affect
LPA	Low Positive Affect
MBHEMR	Mood Based Hybrid Ethiopic Music Recommender
MIR	Music Information Retrieval
MMR Model	Music Mood Recognition Model
P	Pleasantness
RMS energy	Root Mean Square energy
Std	Standard Deviation
UnP	Unpleasantness
ZCR	Zero Crossing Rate

List of Algorithms

List of Figures

List of Tables

Chapter 1: Introduction

1.1 Background

Music is the very ubiquitous phenomenon playing an important role in many people's daily life. People compose, perform, listen to music in everyday lives for many purposes, ranging from aesthetic pleasure, religious or ceremonial purposes, or as an entertainment product for the marketplace [1].

It has been a long time since music's benefit for mental health is known [2]. In the work of Selam Taddesse [1], it is stated that ancient philosophers such as Aristotle and Plato used the power of music for mood/emotional catharsis in their subjects and as a medicine of the soul. King David also played a song for King Saul to ward off the depression that he used to suffer from. Modern researchers also proved that music can change our mood at any time whether we are in a workout, traveling or just relaxing [3, 4]. Music's purpose in our daily life is increasing. Patients who are suffering from burns and cancer are getting important treatment with the help of music [5]. Music utilization in military bands and sporting events is common to stir energy, raise confidence and courage. Schoolchildren use music to remember their ABCs. Shopping malls play music to entice consumers and keep them in the store. Dentists play music to help calm nervous patients.

One of the discovered human behaviors to be affected by music is heart-rate [6]. As mentioned above music also affects a person's mood [5], and heart-rate is responsive to one's mood. So, we can exploit this relationship among music, mood and heart-rate variability to recommend music for a person depending on their mood as relief or boosting positive mood and confidence.

The web permitted us to store as well as distribute contents more easily. Music standout as one of the best captivating web content that attracts the attention of considerably large amount of Internet users worldwide. However, an enormous amount of music is produced and spread through the Internet. This brings its own challenges to find more relevant content. For example, it is ever so challenging to discover more important music for oneself suitable in current mood and context. Nevertheless, some music succeeds in attracting the attention of millions of users, while some others, which are critical for a given user's mood and context remain obscure. But if the whole music is classified into classes of mood and only music from a specific class of music is recommended, it will be easier for the user to discover those obscured and relevant music to their current state of mood instead of searching the whole songs available on the web. In addition to this, the chance of new music to be discovered will increase.

As mentioned above, huge amount of music contents are waving in the Internet to attract more user, in this circumstance recommendation systems should do more work for the user in-order to retrieve and suggest music depending on their actual situation, for instance, mood state (e.g. happy, sad), heart condition (e.g. healthy, abnormal) or any other contextual conditions that might influence the user's perception of music such as location of the user (e.g. city), time related information (e.g. time of the day, day of the week) weather-related information (e.g. sunny, rainy, cold, hot), activity (sporting, studying). Obviously, in a recommendation system, the system does not do everything for the suggestion, the user also contribute by clicking, signup, providing personal profile etc. to play his favorite song or update and modify the list of music to listen. But recommendation system will be more efficient and semantic, if we use more and more personal data about the user for that, one of the ways to get more personal data about the user is automatically collecting data using sensor installed on mobile phone like heart rate sensor and textual data (short messages and social media posts) in addition to contextual data listed above in this paragraph. This also reduces user's effort in providing information.

To meet users' demands for a recommender system, there are some music streaming websites already providing music recommendation services, for example, Spotify[1], Pandora[2], Beats Music[3], AddisZefen[4], etc. The way how they compile their recommendation list varies between companies. Some websites make up recommendations based on users' listening records; some recommend the music that the "neighbor user" listens to, which means that the system assumes that they share a similar taste. Although there are already lots of different ways to draw up recommendations, users are still not satisfied with the recommendation service [7]. One of the drawbacks in the recommender system, mentioned above, is that the systems compile the recommendations based on data of "neighbor user's" taste instead of the user's current state of mood.

[1] https://www.spotify.com/
[2] https://www.pandora.com/
[3] http://beatsmusic.com/
[4] http://addiszefen.com/

It is well known that music mood perception is influenced by cultural factors, such as listeners' acculturation or familiarity with musical background or language [8]. The sites listed in the above paragraph (Spotify, Pandora, Beats music) are mostly focused on the western music which does not fit with the music mood perception of the Ethiopian listeners which have different culture and language. Even though, AddisZefen and other local music streaming websites present Ethiopic music, the way they recommend music does not consider user's contextual situation like mood.

Other than worldwide popular songs, Ethiopians also are interested in their own unique and extremely diverse music. Each ethnic group of the country has it's own associated music and scales (Kiñit), from two-tone scale to complicated diatonic scale [9]. The different type of pentatonic scales are among the prevailing scales (Kiñit) utilized in the county. These and the other commonly used scales include Tizita, Batti, Ambassel, Anchihoye, Dorian pentatonic (Ymatebela wofe), Phrygian pentatonic (Shegiye Shegiye), Mixolydian pentatonic (Ieyew Demamo) etc. [10].

Those sites mentioned above also use static user's mood (always the same type of mood) as a parameter for the recommendation, this does not consider changes in user's mood. In the actual world users mood is not always the same, it might change through time [11].

This work is aimed at exploiting already available data on user's smartphone. Text, biometric data (e.g. heart rate) and music listening experience are some of those data that can be easily found on a smartphone without requiring additional user's effort. These data then used to detect user's context (mood) and make mood based music recommendation.

In order to recommend suitable music for a given user mood, the music on hand should be classified into different types of mood (happy, sad, relaxed, excited, active, warmhearted, blue, fearful etc.). Therefore, this research also proposed music features and algorithms that can best classify Ethiopic music into different type of mood from the audio music.

1.2 Motivation

There are lots of reasons that motivate us to raise the idea of this research. The absence of mood based music recommendation system, considering Ethiopic music, is the first reason. Even though recommendation system is helpful to alleviate the problem of information overload there is no such system for Ethiopic music available publicly.

The other reason is lack of systems to organized and categorized Ethiopic music into classes of mood. People need for a music is often related to the mood in a specific situation. However, to satisfy this need, we have to make a hard search. This is because current music streaming websites classify Ethiopic music by artist name, album or language, they don't consider mood. Some of the western music that are classified into classes of mood does not fit with Ethiopian cultural context, e.g. some systems can classify slow or fast songs [12]. Because human's perception of music mood differs from culture to culture [8].

Music has been among the primary domains for research on recommender systems, starting from early in 1994 [11, 13]. Moreover, people have considered that music is an important aspect of their lives and they listen to music as a frequent activity [3, 4]. This has created new demand for services that support music navigation, discovery, sharing, and formation of user communities. The music recommendation system could contribute a lot for this demand. However, many research questions are still open in order to make the recommendation system more satisfying. Due to this fact, it is interesting and relevant to work on this research domain considering smartphone into account. Because smartphones are one of the main devices used to listen music and also help to track user experience and context.

1.3 Statement of the Problem

Previous researches [14, 15, 16, 17] carried out on music recommendation exhibited satisfying to the user. However, there are still open questions that need additional research, which are stated as follows.

In most of music recommender systems, users' mood is not taken into account. Even some of the system which tries to consider user mood, assume the user mood is always the same unless the user changed his profile settings [18]. Therefore, in order to choose best songs based on current mood, the user has to manually search through a large set of songs. This kind of manually browsing of songs and creating an appropriate playlist based on an individual mood is a very tedious, time-consuming, labor-intensive and upheld task.

Ethiopia is musically a traditional country. Of course, popular music is played, recorded and listened to, but most musicians also sing traditional songs, and most audiences choose to listen to both popular and traditional styles [9]. One of the tasks in a recommendation system is classification of music into different categories of mood. To our knowledge, there is no previous study which attempts to classify Ethiopic music based on mood.

Today's recommender system favors popular songs which have many ratings that are likely to be recommended and will receive more ratings. As a result, the popular songs overwhelm new or unpopular songs, so the novelty and personalization would be weakened [7]. This sort of recommendation based on content properties and collaborative filtering are not sufficient. Such recommendations lack awareness of the contextual situation of the user such as user's physiological condition (heart-rate), activities and other temporal contextual information like mood. To alleviate these drawbacks there should be a new mechanism that will consider interestingness of the contents and semantic attributes of the music (e.g. mood) to fit with user current state of mood. Previous studies [19] suggested that by exploring important features that correlate with music mood, useful recommendations may be produced even before a music becomes popular.

Even though there are already existing data collected by technologies installed on smartphones (e.g. heart-rate sensor) they are not used to personalize music recommendation. Such data are important temporal information about the user that support determining user current state. Integrating these data will reduce efforts of the user.

Some of previously proposed music recommender systems depend on a single type of context information such as environment, location etc, [11]. Recommendation based on such a generic information might lead to wrong decision because the generic data may have exceptions. Incorporating numerous sources of user context information (e.g. text and physiological data) reduces uncertainty on recommendation and increases confidence in the system.

1.4 Objectives

General objective

The primary objective of this research is to design and implement music recommendation system on a smartphone that aware user mood related to heart-rate and other contextual data.

Specific objectives

To achieve the general objective of this research the following specific objectives are identified:

- Study formation and structure of Ethiopic music system
- Study the different music features that represent the mood

- Assess techniques and approaches to extract important information from music
- Extract music features automatically and develop a model that categorizes music based on mood
- Develop an approach that associate heart rate and textual context with user mood in order to recommend music
- Identify techniques to associate music mood with user perception of music
- Collect and prepare music corpus, sensor and text data for training and testing purpose
- Design and implement a prototype of mood based context aware music recommendation system that consider user context
- Test and evaluate the capability of the prototype

1.5 Methods

The activities to be carried out in order to achieve the objectives are presented as follows:

Literature Review: literature review is conducted to survey books, scholarly articles, and any other sources relevant to mood based music recommender.

Data Collection: in order to come up with a system that would be most beneficial for the user, music and the user-related information is collected in addition to previously available data related to music recommendation system studies.

Music dataset which contains the audio and song track information is prepared for training and testing purpose. The different sources of data include Ethiopian music streaming websites, social media and others. Music mood data is collected from the users using a survey (both offline and online) that request manual annotation music.

Design and Implementation of Prototype: prototype of proposed system is development in-order to experiment with the proposed solution.

Tools and Techniques: in order to implement and experiment with the proposed solution, several tools are used. Audacity, MATLAB, MIRtoolbox, and Python are among the mainly used tools.

Evaluation: in order to evaluate and test the proposed system, we have used different evaluation methods for individual components and overall system.

Music mood classification: there is no previous work related to music mood classification for Ethiopic music. So, the performance of the proposed music mood classification method is evaluated by comparing the performance of other commonly known algorithms applied to the same set of music and audio features.

User mood detection: performance of the user mood detection component is evaluated by comparing its accuracy with best results of previous similar studies.

Recommendation: in the preliminary study we didn't find similar research to compare our system. Therefore, we performed an experiment on people to evaluate the system if they agree with the established match between a song and mood.

1.6 Scope and Limitations

The scope of this study is focused on designing and implementing mood based context-aware music recommendation system for smartphone by identifying user's perception towards Ethiopic music mood. Automatically classifying Ethiopic music into different classes of mood, based on the mood factor from the audio signal; and detecting user's mood from contextual information. Then creating an association between user's mood and distinct types of music mood. This system is aimed at providing useful listening experience for the user, provide good insights for music producers about listeners perception of Ethiopic music mood, and enhancing music streaming websites recommendation ability.

The proposed research has the following limitations:

- The music mood model used in this research is prepared in English language which is not used in daily activities of the public. This is because, currently there is no music mood model available considering Ethiopian culture and language.
- The query sentences used to train the system are English sentences because we didn't found Amharic mood sentence which fits for this study.
- Even though music mood can be reflected in different parameters such as audio, lyrics, artists etc., due to limitation in time and data issues, this study is limited only audio features.

1.7 Application of Results

The outcome of this study is expected to be one of the contributing research for the following purpose:

Music Streaming Websites: music streaming websites can use the system to recommend or create a personalized playlist or update playlist based on the mood of the listener or contextual situation of the user. The system also could be used automatically classify music database content into different categories of mood. This allows for the streaming websites to present labeled/categorized music based on mood.

Media Players: intelligent media players could take the user mood as an additional parameter to generate or modify existing playlist. The proposed system also enables for media players to learn about the situation and use this information to recommend music.

Music Information Retrieval (MIR) Systems: music classification is one of the main tasks of MIR. The techniques used here to classify music can also be used in the MIR systems.

Systems Involving User Mood Detection: one of the tasks in the proposed system is detecting user mood for the purpose of music recommendation. The techniques used in this research to detect user mood can be applied to other system involving user mood detection.

1.8 Thesis Organization

The rest of this thesis is organized as follows, Chapter 2 presents a literature review on recommendation systems focusing on music recommenders; Overview of Ethiopic music and the relationship between music and mood. Chapter 3 discusses the approaches, techniques, and tools available and used by the researchers related to music recommendation system. Chapter 4 shows the design of mood based context-aware music recommender. Chapter 5 discusses experimental results of the research. Finally, in Chapter 6, conclusion and future works are presented.

Chapter 2: Literature Review

This literature review concerned on music recommendation systems. There are four main approaches of mood based music recommendation. This chapter discusses generally recommender systems; and then about each approaches of music recommender systems. These also include components of the recommendation system, algorithms, available techniques and tools, music mood models, the status of music and mood-related studies in Ethiopia.

2.1 Recommendation System

Recommendation system (engine or program) help narrow choices to those that best meet users' particular needs [20]. It is a system which attempts to predict information/items that a user may be interested in. These systems often use both, query supported retrieval approaches and information filtering approaches based on information about the user's profile.

A recommender system is part of information filtering system that tries to predict user's responses to options/items (e.g. music, movies, books, news, web pages or people) from big data [20]. There are different ways in which most of the recommender engines produce a list of recommendations. Collaborative Filtering (CF) approach, Content-Based Filtering (CBF) approach, Context-Aware Filtering (CAF) approach, and Hybrid approach are among the major ones.

CF approaches build a model from a client's past experience and comparable choices made by different clients. This model is then used to anticipate things that the client may have an enthusiasm for [21]. CBF approaches utilize properties of items for an extra recommendation of items with comparable features [21]. CBF approaches utilize a series of discrete characteristics of an item in order to recommend additional items with similar properties [22]. CAF approaches generate relevant recommendations by adapting them to the specific contextual situation of the user [23]. The combination of these approaches (Hybrid filtering approach) is also applied often to make the recommendation more effective in some cases [24]. Hybrid approaches can be implemented in several ways: by making CF and CBF predictions in a CAF system; by applying the CAF approach in CBF or CF approach; or by unifying the approaches into one model for a complete review of recommender systems.

2.2 Music Recommendation System

Like recommendation systems of the other domains, music recommendation system uses similar approaches such as CF, CBF, CAF techniques and/or these methodologies are joined together to make more satisfying and quality recommendation services.

Unlike the items/information to be recommended in other domains, music suggestion is a challenging work due to its multifaceted nature of content as well as how humans perceive music, which is still not fully comprehended. Music recommendation is a complex series of tasks that can be influenced by familiarity with a music listening background, socio-economic, cultural background, and many other factors [8]. Here some of the more common general techniques of music recommender systems are reviewed. These techniques consider contextual status, historical ratings and music listening habit of the user combined with music features that could affect users' preference which includes low-level features (e.g. rhythm, pitch) and high-level features (e.g. genre, artist, instrument).

2.2.1 Components of Music Recommendation System

A music recommendation system has different components that work together to predict user's interest in music. These components could be classified into different classes depending on the specific purpose. Yading et al [20] in their survey of music recommendation system, generally classified these into three main components. These are User Modeling, Music Item Modeling, and User-Item Matching Algorithms.

User-Modeling Component: collects vital information about the user to process prediction of user's music taste. The component has two elements, User profile modeling and User listening experience modeling. The facts about the user could be acquired by two ways using an explicit model (built based on information provided by the user) and implicit model (built based on information collected by the system itself by observing users experience with the system).

Item-Modeling Component: makes use of a chain of music data to find characteristics (Metadata). Pachet [25] labeled the music metadata into three categories: editorial metadata, cultural metadata, and acoustic metadata. Editorial metadata (album cover, composer, artist name, genre etc.) obtained by means of an expert or a group of experts. Acoustic metadata (tempo, pitch etc.) obtained from an evaluation of the audio signal.

Cultural metadata (similarity between song items) acquired from the evaluation of corpora of textual data of songs.

User-Item Modeling Component: associates user with their preference of items in different recommendation approaches (collaborative filtering, content-based filtering and others).

2.2.2 Music Recommendation Approaches

Metadata Filtering Approach: Metadata Filtering Approach is fast and accurate. This approach utilizes editorial metadata of the music provided by experts like lyrics, artist name, album cover, song title etc. to search songs [26]. Metadata Filtering approach is an easy and quick way of searching; furthermore, its results are exact. However, it requires the user to know about editorial metadata of songs he/she is looking for. In addition, it takes a long time to maintain expanding metadata. Moreover, those suggestion given by the system is moderately poor. Since its recommendation is based on metadata, users feeling is not considered.

Collaborative Filtering Approach: CF approach works on connected numerous music information. This system depends on client generated content or certain input, and the "word of mouth" way to deal with suggestions of music that are liked by comparative clients. Hence, CF doesn't have to manage every detail of the content, i.e., they don't base the recommendation on the physical properties of the song or its description. There are three types of CF approaches, Memory-Based, Model-Based, and Hybrid approaches.

Memory-Based: this technique makes use of client rating data to compute the similarity between the clients or items to make recommendations. Let U be the set of all users, and I is the set of all items. Then the rating data is stored in a matrix R of dimensions $|U| \times |I|$ depicted in Figure 1. Where $r_{u,i}$ in a row u is the rating of user u gave to item i, or is empty if the item has no rating or not known.

CF systems predict ratings for items whose rating is not known. An unknown rating of user u for item i can be predicted either by finding a set of users similar to u (user-based collaborative filtering), or a set of items similar to i (item-based collaborative filtering), and then aggregating the ratings of similar users/items.

The following formula (Equation 1) is commonly used for user-based collaborative filtering [27].

$$\hat{r_{ui}} = r_u + k \sum_{v=1}^{n} w(u,v)(r_{vi} - r_v) \quad \textbf{\textit{(1)}}$$

Where

- r_u is the average rating of user u
- n is the number of users with known ratings for item i
- $w(u, v)$ is the similarity of users u and v
- k is a normalization factor such that the sum of w(u, v) is 1

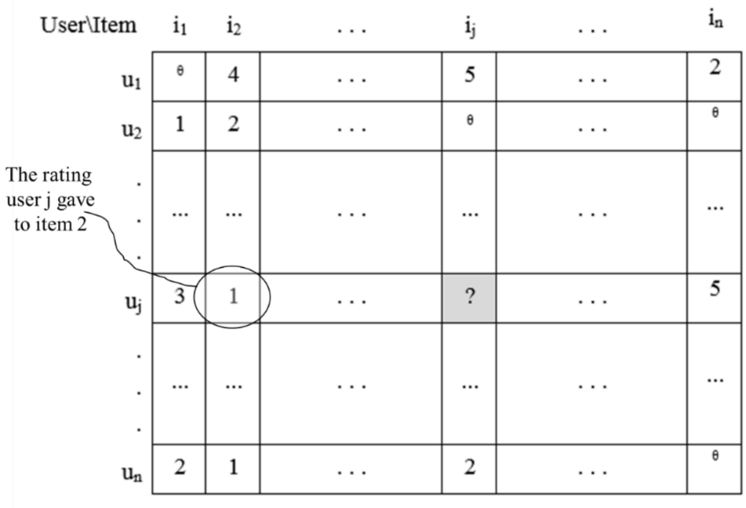


***Figure 1**: User\Item Rating Matrix*

Implementation of memory-based collaborative recommender is comparatively easy. Moreover, new data can be added easily and incrementally as well as it is not expected to consider the content of the items to be recommended. However, since it is dependent on users' ratings, the performance of the system decreases when user data are sparse. It is also difficult to recommend new users and items. Scalability for large datasets is another issue.

Model-Based: models are created by using data mining techniques, and the system learns algorithms to look for trends using training dataset of user ratings. These models are then used to come up with predictions for new songs.

Here is a formula (Equation 2) commonly used in model-based collaborative filtering [27].

$$\hat{r_{ui}} = \sum_{j=0}^{m} Pr\left(r_{ui} = j | r_{uk,} k \epsilon R_u\right) j \qquad (2)$$

Where

- R_u is the set of ratings of the user u between 0 and m
- $\Pr(r_{u,i} = j \mid r_{u,k}, k \in R_u)$ is the probability of rating j given by user u for item i

Model-based approach better reduces the problems of memory-based approaches such as the sparse of user information and scalability issues. It also improved the prediction performance for recommendations. But this method also has its own drawbacks, such as, expensive model-building, and lose of useful information for dimensionality reduction techniques.

Hybrid: in order to use advantages of both memory-based and model-based CF, various programs combine the algorithms.

Content-Based Filtering Approach: CBF recommendations in music domain have been used considerably less. The reason for this might be that content-based techniques require knowledge about the data, and music is notoriously difficult to describe and classify. CBF approaches typically exploit traditional music information retrieval techniques like an acoustic fingerprint or genre detection.

Content-based recommender store describing data of the songs to recommend similar songs to those known to be liked by the user. Songs are commonly represented by n-dimensional feature vectors. Those Characteristics describing the song could be gathered automatically or manually.

One of the main tasks in CBF approach is learning the user model in view of client inclination. Relevance Feedback and Nearest Neighbor approaches are typical instances of content-based approach. The Nearest Neighbor algorithm saves information of the items that are evaluated by the user, either implicitly or explicitly. Depending on this stored information, class of a new item is decided from labels of nearest neighbors. Relevance feedback (Equation 3) applied in training the user's profile vector. The user profile vector is empty at the begging, and it gets updated iteratively until the user profile vector is enough to represent user's preferences [28].

$$q_m = \alpha q_0 + \left(\beta \frac{1}{|D_r|} \sum_{d_j \in D_r} d_j\right) - \left(\gamma \frac{1}{|D_{nr}|} \sum_{d_j \in D_{nr}} d_j\right) \qquad (3)$$

Where

- q_m is the modified vector
- q_o is the original vector
- D_r and D_{nr} are the set of relevant and non-relevant items

- α, β, and γ are weights that are shifting the modified vector in a direction closer, or farther away from the original vector

Context-Based Filtering Approach: context-aware recommender systems create more pertinent suggestions by adjusting them to the particular logical circumstance of the client. The data showing user situation could be any data that can be utilized to describe circumstance of the user. Dey [29] defined the context in computing systems as "any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and application themselves". Mood, environmental conditions, time, and current activity are few examples of contexts in recommender system domain.

Now day's users get information and services in various environment and circumstances due to advances in mobile computing. Smart devices and wearable computers open doors for researchers to gather and use relevant information to advance the connection between users and computer. Contextual data like the mood of the user, physical environment, time, events can help the recommender system to better comprehend the present needs of the client. Despite the fact that there have been researches on utilizing contexts in recommender system [30], applying CAF techniques in the domain of music recommendation system is still little explored.

There are few categorizations of contextual data presented in different studies. The simpler categorization incorporates user-related context, environment-related context and multimedia contexts that are information the user is exposed to besides music, e.g., text, images.

User-related context includes the activity of the user, demographic information of the user, and emotional state of the user. The activity of the user includes actions such as walking, reading, running etc. or numerical data that could define user's state like heart-rate. Such type of context can have an immediate effect on human's musical choice. For instance, individuals favored distinctive musical rhythm contingent upon their occupation [11]. Demographical information of the user is user related contexts such as age, gender, personality traits, etc. Studies in music psychology show that these type of contexts has a direct effect on user's music preferences [31]. The feeling of the user has a direct influence on user's musical preferences. For instance, we all listen to very different music in a sad mood compared to when being happy [32]. One may prefer to listen to totally different music for a tragic situation contrasted with happy mood. Music sometimes portrayed by mood due to the fact that it incorporates moods in it.

Environmental context includes the location of the user, time, weather condition, and other facts that contribute to the emotional state of the user. There is a relationship between affective qualities of the music listening environmental conditions and the choice of music that fits with these conditions [33]. This relation motivated many researchers to consider the influence of environmental context for music recommendation.

Hybrid Filtering Approach: hybrid recommendation approach consolidates advantage of CF, CBF and CAF approaches, in-order to reduce drawbacks of the individual methods. This method helps to avoid the issue of new item or users in the CF system and modeling new users issue in CBF recommender systems.

There are different strategies to consolidate different recommendation techniques [34], these includes:

Weighted Score Method: in this technique different recommendations are transformed into one by processing and joining the scores provided by the different methods.

Switching Method: the system switches between distinctive methods depending on the characteristics of the dataset, quality of recommendation results or other certain criteria.

Mixed Method: recommendations from different techniques are presented together either side by side or in a combined list.

Feature Combination Method: item features that work well with specific recommendation methods are brought together into a single recommendation algorithm.

Cascading Method: the result of one recommendation method is refined by another method. For instance, CF approach could be used to rank suggested items, afterward, CBF can be applied to modify the recommendation.

Feature Augmentation Method: this technique uses outputs of one recommendation method yielded by another recommendation approach as an input.

2.3 Mood and Music

Music puts the listener in a certain state of mood or feeling. Automatically detecting the mood of a music that can be perceived by the user is a challenging task. This is sometimes related to the field of music psychology to some extent. Because it involves dealing with the human impression of a given song and music mood modeling [35].

Automatic music mood detection can be applied in music recommendation system to predict likable songs. The following paragraphs discuss psychology-based music mood models and machine learning techniques to detect mood in music content.

2.3.1 Mood Models for Music Cognition

The mood in the field of psychology is defined as a longer experience without specific object connection. While emotion is a short experience in response to an object [36]. However, music information retrieval or music mood recognition, researchers use the terms mood and emotion interchangeably [11].

Defining the possible set of moods conveyed by a song is a preliminary task in the automatic music mood detection processes. Human moods are complex and multi-layered. Different aspects of primary mood create other secondary lists of moods. A number of researches are conducted in cognitive psychology to come up with universal mood model. But there is no universal taxonomy of mood model that has been agreed on. Therefore, mood model is chosen based on the task and domain of the research [37]. Music mood models are mainly conceptualized as categorical models and dimensional models [38].

Categorical Models: moods are recognized with the help of adjectives denoting moods or class tags. There exist a limited number of mood classes, from which all other secondary moods are derived [39]. The categorical model either makes use of six basic mood classes namely anger, disgust, fear, joy, sadness and surprise [16] or uses domain-specific expressive classes.

Many researchers of affective computing have proposed mood models for describing affect states of a human. Example include, the Hevner's 8 clusters of affective terms (Figure 2) which is latter regrouped into 10 clusters by Farnsworth [40], and into 9 by Schubert. The Tellegen-Watson-Clark model with 8 adjective groups (Figure 3) [41] is also another common example of categorical mood model.

Dimensional Models: allows to represent mood states as continuous values on independent two or three dimensions. Each mood is a location in a multi-dimensional plane, based on a reduced number of axes (2D or 3D). Although there have been attempts to use the three dimensions (Valence, arousal, and potency) [42], the most common models that are used in practice include the two dimensions (Valence and arousal) [11, 39]. Examples of the dimensional model include the Circumplex model of James Russell's (Figure 4) [43] and Robert Thayer's model (Figure 5) [44].

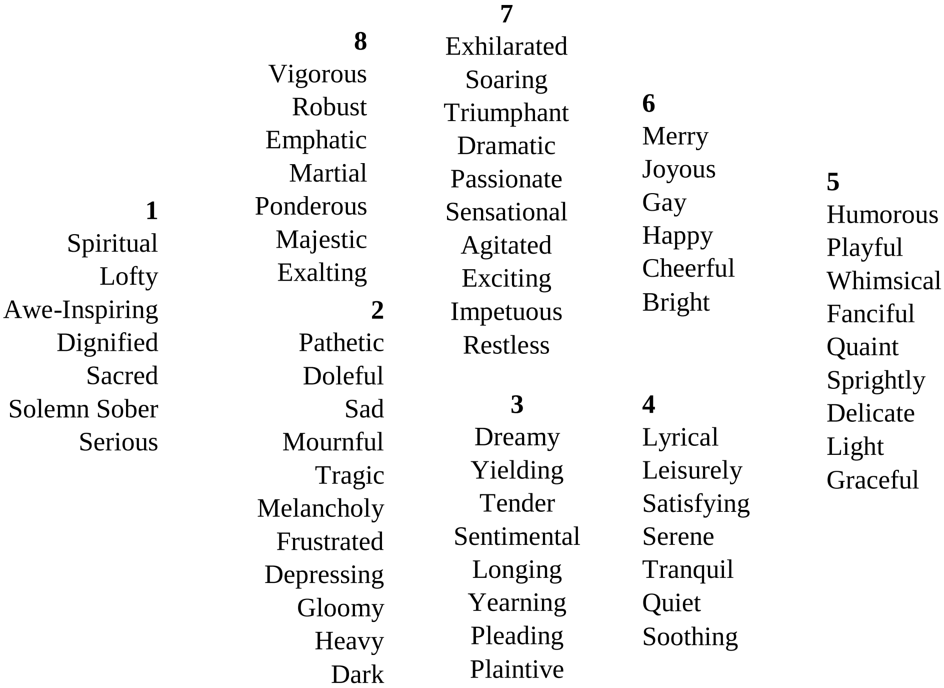


***Figure 2**: Hevner's eight clusters mood model (Hevner "Mood Clock")*

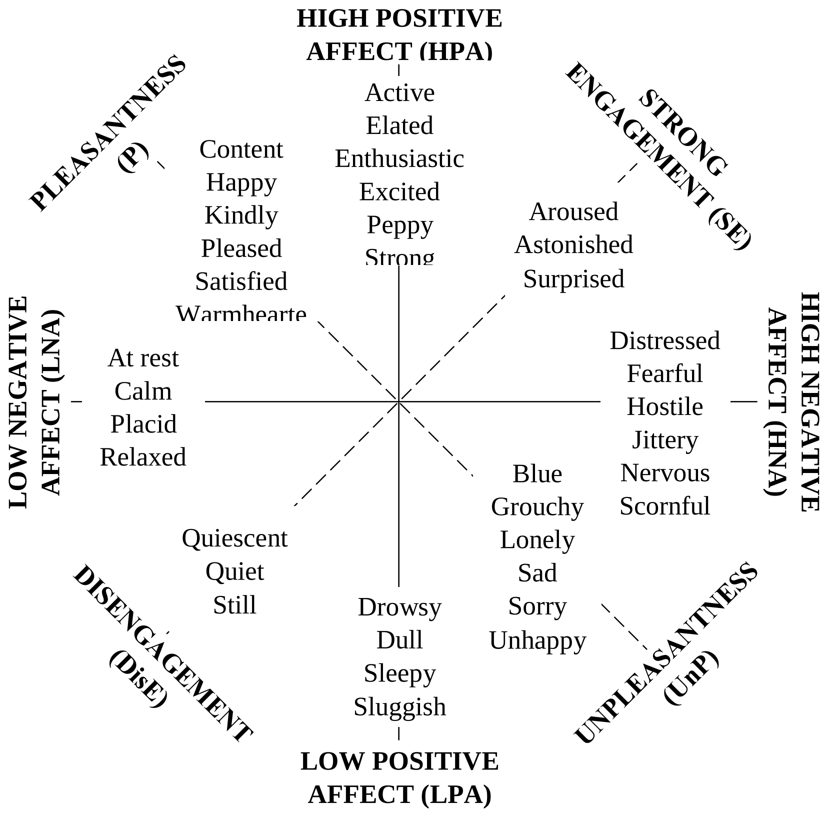


***Figure 3**: Tellegen-Watson-Clark mood model*

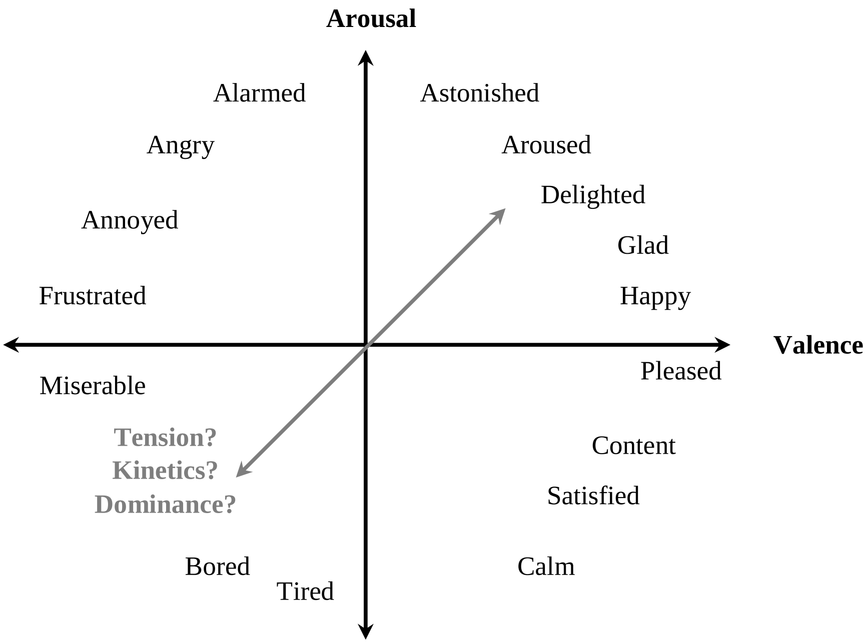


Figure 4*: Circumplex model of James Russell*

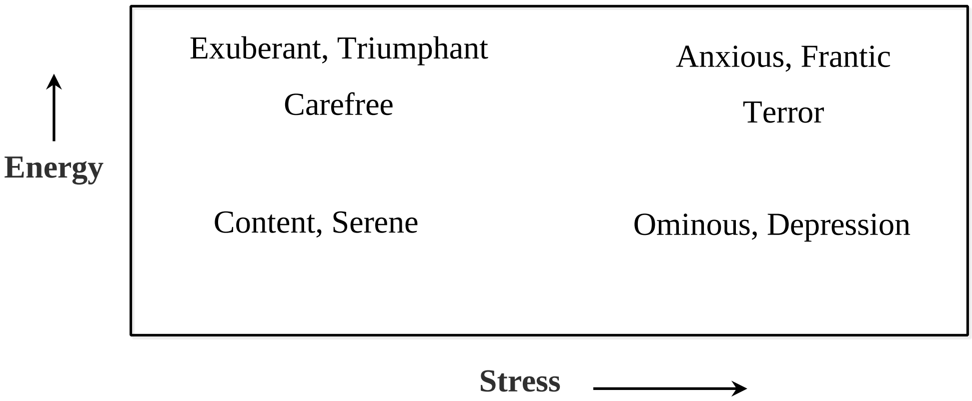


Figure 5*: Robert Thayer's mood model*

2.3.2 Acoustical Analysis of Music

Music in the field of sound engineering is described as the vibration of air molecules that causes regular vibration of the eardrum when it is radiated from the musical instruments [45]. It is recorded through a sampling of the magnitudes of audio vibrations at many points in time. These samples are then analyzed to extract music features that affect human mood [46, 47]. Numerous features are proposed to describe audio signal in music [48]. The most relevant features that are proposed to detect mood in a music include tempo, pitch, tonality, key, harmonics, loudness, MFCC etc. [49, 50, 51].

Intensity is another term for the loudness of a song that is measured by the average volume of a song track. The Root Mean Square (RMS) energy is one of the main elements of music intensity that is used to decide energy of a song. The RMS energy is computed by decomposing the audio (e.g. Figure 6) into frames (e.g. Figure 7). Then the root average of the square of the amplitude of the audio signal (Equation 4) [52, 48].

$$S_{rms} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} a_i^2} = \sqrt{\frac{a_1^2+a_2^2+\cdots+a_n^2}{n}} \qquad \textbf{\textit{(4)}}$$

Where *S* is the audio signal; a is the amplitude of signal *S.*

Figure 8 shows an example of RMS energy of Mahmud Ahmed's Tizita track. Table 1 also presents 6 different RMS energy values of the same track.

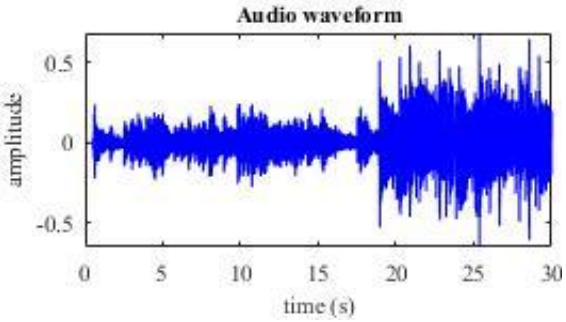


Figure 6: *Waveform of Mahmud Ahmed 'Tizita' track*

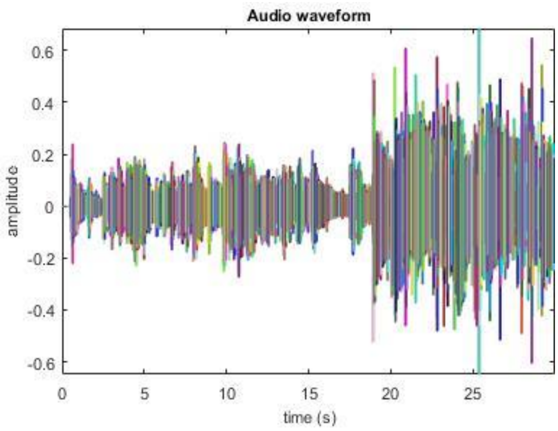


Figure 7*: Frame decomposed audio track 'Tizita'*

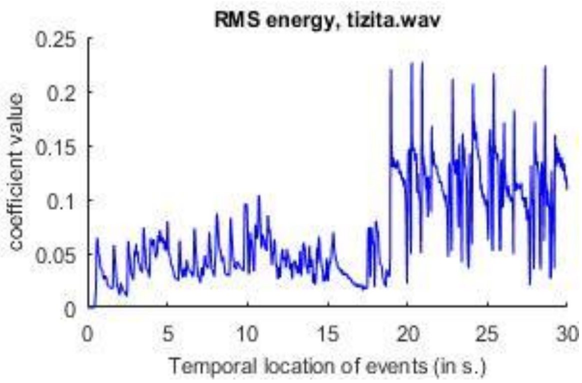


Figure 8: *RMS energy of audio track 'Tizita'*

Table 1*: RMS Energy values of tizita.wav*

RMS Energy	**Values**
Mean	0.114917
Std	0.03692

Low Energy Rate shows the temporal distribution of the energy that helps to see if it remains constant throughout the signal. This can be found by computing amount of frames less than the average energy in percent. E.g. low energy rate of tizita.wav is 0.61968, it shows about 62 percent of the energy distribution is below the average energy which means it has high low energy rate.

Tempo is the speed of the song that can be recognized by extracting a beat spectrum from the song so that computing the frequency of the beats yields the tempo. A beat is characterized as the pulse or regularly repeating event in a song [52]. E.g. the tempo related to file tizita.wav is depicted in Figure 6 along frames whose average is 93.7491 bpm.

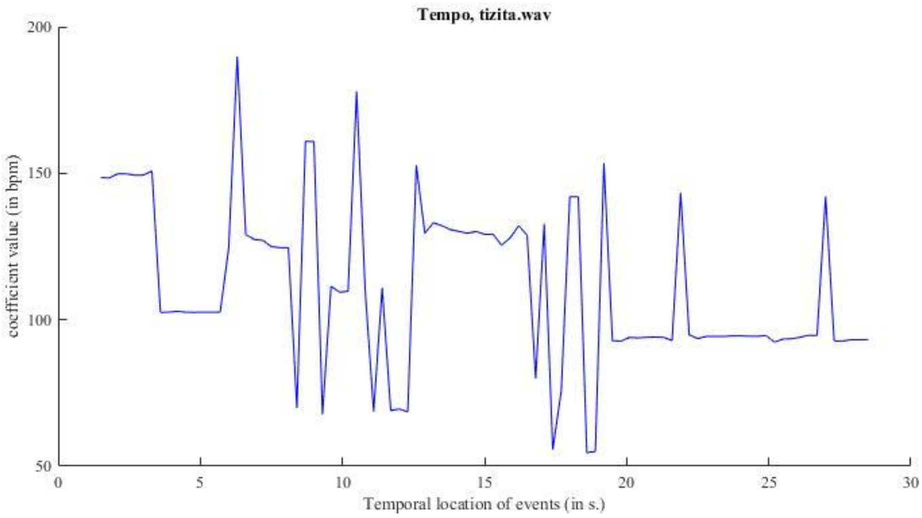


Figure 9*: Event distribution of tizita.wav*

Pitch is the relative highness or lowness of a musical sound. Which can be computed from the frequency of vibration. Fast vibrations make high pitch sound, while slow vibrations make low pith. In the Thayer's [44] mood model, the pitch is used to decide stress level conveyed by a song. Many studies show that higher pitched songs are related to lower stress moods (exuberant, activeness, or happiness) while lower pitched songs related to high-stress moods (sadness, fearfulness or depressed) [52].

Timbre is another term for tone color or the tonal quality of a song. It is the unique characteristics of the musical instrument or voice caused by its harmonic components. Determining timbral difference between two instruments that are playing the same note is relatively easier compared to finding the difference for an entire piece of a song. Zero-crossing rate (ZCR) and spectral irregularity are an instance of timbral elements of a music that can be analyzed from full songs. The ZCR is a good indicator of the noisiness of a signal that is determined by the speed at which a signal crosses the zero line (Figure 10).

Spectral irregularity is the sum of the squares of the difference in amplitude between adjacent peaks. It shows the degree of variation between successive peaks along the frequency spectrum. The higher rate of these two values and other timbral elements (spectral centroid, spectral entropy etc.) shows that the audio signal has a large amount of harmonics. This, in turn, indicates that the song has a high amount of energy [52, 48].

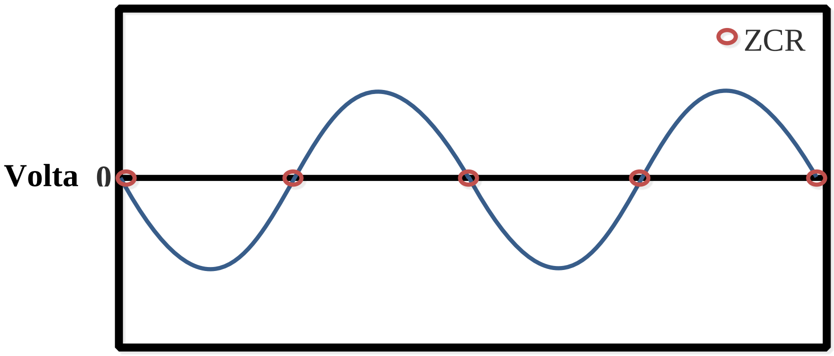


Figure 10: *Zero-crossing rate of a signal*

Music mood analysis also involves using other algorithms such as the fast Fourier transform (FFT). FFT shows the amounts of various frequencies in a signal from the time domain [52].

Fluctuation is a rhythmic periodicity along auditory channels. It is estimated by spectrogram computation transformed by auditory modeling and then a spectrum estimation in each band. First, the power spectrogram is computed. Then an FFT is computed on each band, subsequently, the resulting spectrum across bands is summarized.

Entropy of Spectrum: it is a relative Shannon entropy of spectrum of the audio input that is computed using Equation 5 independent of the sequence length.

$$H(p) = \frac{-\sum plogp}{log(length(p))} \qquad \textbf{\textit{(5)}}$$

Where p is the peak of the audio input curve.

Mode describe the song as major or minor. It is expressed in terms of numeric value between -1 and +1. If a song has a value of mode closer to +1, that song is more major mode or if it has a value of mode closer to -1, the modality of the song is more of minor. E.g. 'tizita.wav' has mode value of -0.05082, this implies that it is Tizita minor.

HCDF is the rate of music chord change related to the notes, which is computed from the flux of the tonal centroid.

Roughness also known as sensory dissonance is estimated by computing the peaks of the spectrum, and taking the average of all the dissonance between all possible pairs of peaks. An instance of roughness using tizita.wav depicted in Figure 11.

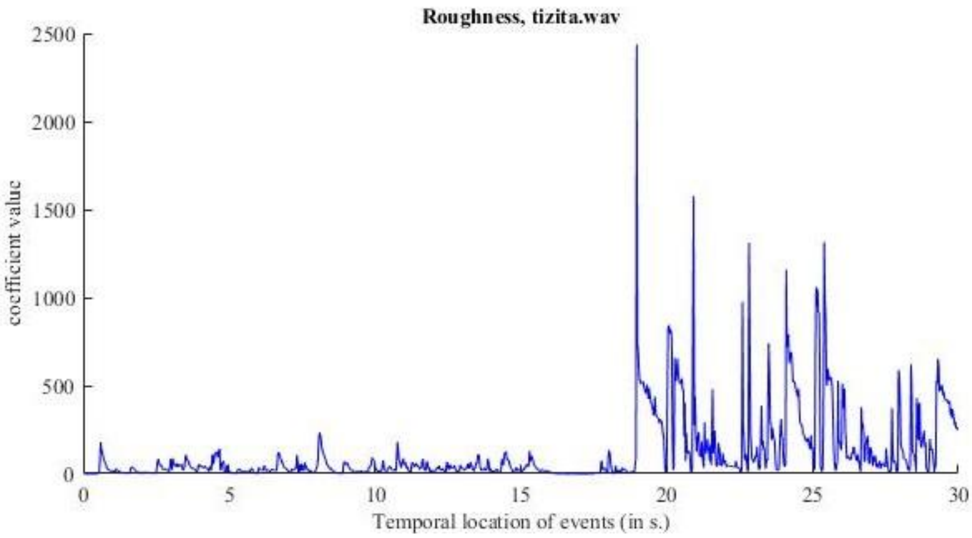


***Figure 11**: Roughness of tizita.wav*

Key clarity is the key strength associated to the best keys (peak abscissa), i.e., the peak ordinates. An example of key clarity is depicted in Figure 12 using tizita.wav file.

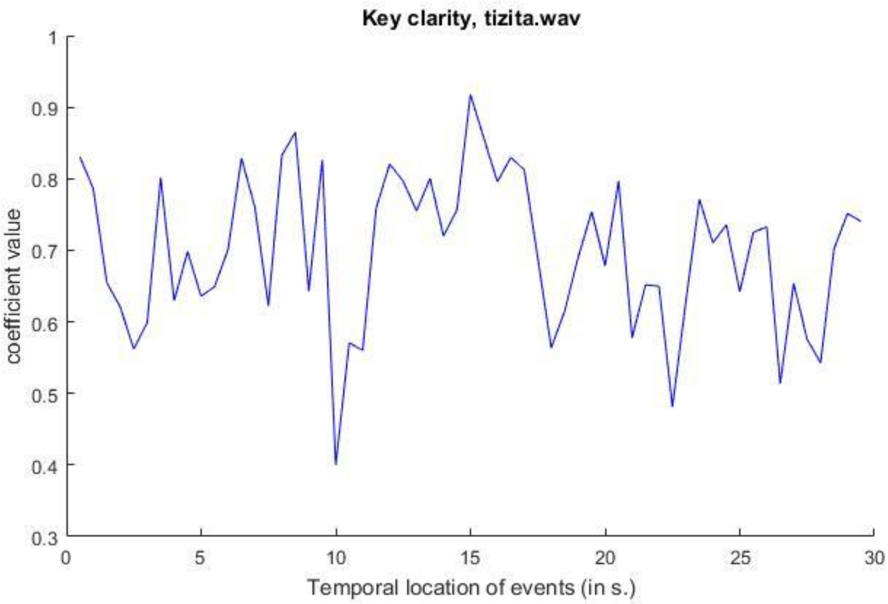


***Figure** 12: Key clarity of tizita.wav*

Attack Time helps in analyzing timber properties of the audio, it's estimated by computing temporal duration of events. An instance of audio attack time of tizita.wav is depicted in Figure 13.

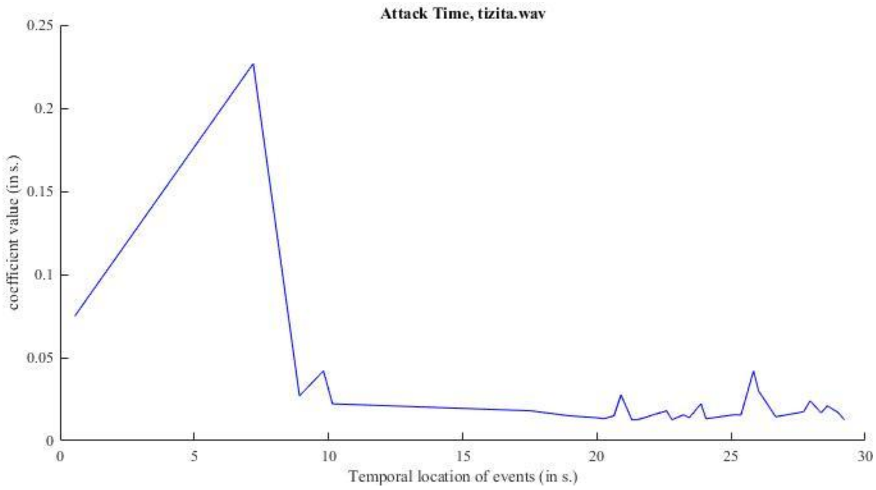


***Figure** 13: Attack time of tizita.wav*

2.3.3 Heart-Rate for Mood Detection in Music Recommender

Several studies have investigated the relationship between mood and heart rate during [53, 54, 55, 56, 57]. Detecting user mood from heart rate variability is easy because it can be measured continuously with a short intervals of time, it does not interrupt the user other activities [58]. It is important to know the relationship between heart-rate and mood. There are number of ways to employ mood information [59]. Music and mood recommender systems such as Netflix or Spotify would benefit from using mood as an input to their recommendation algorithms. By knowing the user's mood and building preferences based on previously selected items, these providers could recommend different media to match the user's current mood. While the system can ask the user to supply their mood, an automatic mood sensor will significantly improve the system's usability.

2.3.4 Machine Learning Approaches to Music Mood Recognition

Many computer science researchers are targeting music mood recognition to find best methods to detect the moods conveyed by a song [60]. Detecting a music mood automatically is important in retrieval, recommendation, and classification of songs. However, the absence of single mood taxonomy and the gap between human perception of songs and the way machine represent the song features makes automatic music mood detection very difficult [37].

Depending on the machine learning algorithms and the set of available moods, the approaches used by the researchers are different. Even though, music mood detection model is proposed as a supervised machine learning task before the set of mood classes known. Achieving more than 60% of precision with a large and varied set of data is a problem [11]. Support Vector Machines (SVM) [38], k-Nearest Neighbors (kNN) [61], Mixed Media Graph (MMG) [62], and Gaussian Mixture Models (GMM) [38] are common examples of machine learning algorithms found in the literature.

2.4 The Dempster-Shafer Theory for Music Recommendation

An attempt to combine different contextual information was made for user modeling in order to drive suitable recommendations. However, the problem of imprecise information, incomplete data, and uncertainty on the available information are major issues that are still not addressed [11]. In the recent years, the theory of belief functions/Dempster-Shafer (DS) theory is becoming a promising technique to alleviate these problems [63, 64, 65].

The DS theory has the capability of dealing with missing or inconsistent data, it incorporates beliefs of different sources to handle unreliable or ambiguous data. Default data or expert view data may have exceptions, the DS theory also allows such data to be supported by additional sources of information [65].

The DS theory is a mathematical approach. Basically, it has two main steps, first, it acquires degrees of belief (also known as mass) in the range of 0 and 1 for one event from probabilities of related events. Then combine the acquired degree of beliefs if they are from different sources.

Belief degree: the DS theory problem formalization begins with identifying the set of all possible hypothesis or set of possible conclusions to be drawn. Let X be the set of all hypotheses under consideration, e.g. $X = \{x_1, x_2, x_3\}$. The power set 2^X also referred to us frame of discernment, is the set of all possible subsets of X, including the empty set represented by ϕ, e.g. $2^X = \{\{\phi\}, \{x_1\}, \{x_1, x_2\}, \{x_1, x_3\}, \{x_2\}, \{x_2, x_3\}, \{x_3\}, X\}$. A basic belief assignment (BBA) function (Equation 6) of the DS theory assigns a belief mass for all members of the power set. Given the following conditions satisfied:

- The mass of the empty set must be zero (Equation 7)
- Sum of all member of the frame of discernment must be 1 (Equation 8)

$$m: 2^x \rightarrow [0,1] \qquad \textbf{\textit{(6)}}$$

$$m(\phi) = 0 \qquad \textbf{\textit{(7)}}$$

$$\sum_{A \in 2^x} m(A) = 1 \qquad \textbf{\textit{(8)}}$$

Here A is a member of the power set 2^X and $m(A)$ is the mass assigned to A that does not give additional clue about subsets of A, so that all the elements of A will have their own mass.

Belief Combination: Dempster's combination rule (equation 10) is a data fusion operator that allows to combine alternative beliefs from another source of evidence for the same problem. Let *m1* and *m2* represents belief functions for different sources of evidences for different hypotheses. The Dempster's rule for combining the two belief functions (*m1* and *m2*) to generate combined belief degree (also known as joint mass) function is computed as follows in Equation 9 and Equation 10:

$$m_{1,2}(\phi) = 0 \tag{9}$$

$$m_{1,2}(A) = (m_1 \oplus m_2)(A) = \frac{1}{1-K} \sum_{B \cap C = A \neq \emptyset} m_1(B)\, m_2(C) \tag{10}$$

Where K (Equation 11) is a measure of the amount of conflict between the two belief functions.

$$K = \sum_{B \cap C = \emptyset} m_1(B) m_2(C) \tag{11}$$

2.5 Music of Ethiopia

The pioneer of music notation St. Yared and other composers made Ethiopia rich in cultural music mosaic. The combination of moods (sorrow, pain, and joy) was one of the repeatedly occurring themes in the history of St. Yared life. That is reflected in the sacred songs composed by him [66].

Music of Ethiopia is mainly based on pentatonic scale (Kiñits), which is characterized by the following features as presented by Ezra Abate [10]:

- A tone system with specified interval structure e.g. Kiñits (Tizita, Ambassel etc.)
- Textual or Vocal music e.g. Azmari music (Griots)
- The melodic structure is based on melismatic or ornaments
- It has a variety of rhythmic patterns that accompany both vocal and instrumental music
- A distinct type of songs for different social contexts e.g. wedding, work, etc.
- A clear difference between sacred and secular music

Ethiopic music also has distinct types of styles that are classified into different manners, such as modes, season, singing techniques or others. Zenebe [67] classified Ethiopic music into four main types of song styles. These are Zefen, Ingurguro, Mezmur, and Zema. Zefen is characterized by its irregularity of meter as isometric and heteromeric patterns. Ingurguro style is characterized by the high falsetto vocal style of female singers (e.g. Aster Awoke singing style combined with Zefen) and deep vocal range for male (e.g. Kassa Tesema's Kirrar[5] songs). Mezmur is sung in full voice with rhythmical restrictions and limitation of contents. Zema related to religious songs, specifically sung for the purpose of adoration or prayer.

2.6 Summary

Music recommendation system predicts user's preferences to narrow choices. Unlike other item recommendation systems, music recommendation is a complex and challenging task due to its content as well as the human perception of the music.

Music recommendation system has three core components, these include user modeling, music item modeling, and user-item matching algorithms. User modeling is vital for user's preference prediction and it is further divided into user profile modeling and user listening experience modeling. Item modeling component characterizes the music using editorial, cultural and acoustic metadata. User-Item modeling algorithms associates user with their preference of songs.

There are list of music recommendation approaches, such as metadata, collaborative, content-based, context-aware, and hybrid filtering. Metadata filtering approach uses editorial metadata of the music. It is fast and accurate, but it doesn't consider user feeling. Moreover, the user has to know about editorial metadata of the song. CF doesn't depend on the physical properties of the song or its description. It has three types of filtering approaches, memory-based, model-based, and hybrid approaches. Implementing CF and adding new data is comparatively easy. However, scalability and new user are major issues. CBF approach, unlike the CF, store a physical description of the music to recommend songs. CAF approaches adjust itself to the particular situation of the client. It uses user-related, environment-related and/or multimedia contexts. Hybrid

[5] Songs played using Kirrar. Kirrar is a string music instrument from Eritrea and Ethiopia

recommendation approach consolidates advantage of the other filtering approaches to reduce drawbacks of the individual methods.

Automatic music mood detection is applied in music recommendation system to predict a list of songs that are relevant to the user current mood. Categorical and dimensional models are the two main conceptualization of music mood models to define the possible set of moods conveyed by a song in the automatic music mood detection processes. Different music mood detection approaches are proposed depending on machine learning algorithms and music mood models used.

Despite an attempt is made to combine different contextual information for user modeling. Imprecise information, and incomplete data. However, DS theory is giving a promising result in other applications other than music recommendation.

Chapter 3: Related Work

This chapter highlights a review of previous research works in the domain of music recommender system. The review outlines different approaches, techniques, data and tool used, evaluation methods together with the results, and lessons learned from strengths and weaknesses of the previous works. Generally, previous researchers used CF, CBF, CAF and others. In this work, we review some of these works as follows.

3.1 Collaborative Filtering Approach

Music recommendation using the CF approach started early in 1994 using email exchange [11, 68]. Hayes and Cunningham [69] proposed Smart-Radio, which is web-based client-server application based on audio streaming technology. The system allows the users to create, customize and share playlists. They computed users' similarity using Pearson correlation based on the ratings previously made. Then users are recommended radio programs using top n neighbors/similar users as the source of recommendations. However, the authors of this paper have not indicated the data they used in their experiment. The evaluation method and results of the evaluation are not given.

Kuzelewska and Ducki [70] proposed user-based and item-based hybrid collaborative music recommendation system. The system is a web-based application that based its recommendation on user similarity (user-based) and artist similarity (item-based). The authors compared different similarity measures to find the best performing algorithm. The Manhattan, Cosine measure, Pear-son and Spearman correlation, Tanimoto coefficient, and Euclidean distance were the similarity measures they experimented with.

They used data from Last.fm[6] that contains ratings of 500 users and 13680 songs of 4436 artists. The Manhattan measure was the best performing algorithm with RMSE of 1.35 and MAE of 0.97. The main drawback of this research was users' preference of songs is determined by others rather than their own perception.

3.2 Content-Based Filtering Approach

Most of the CBF approaches use techniques in the field of information retrieval [11, 71]. Hoashi *et al*. [72] are among the researcher who uses music information retrieval method

[6] https://www.last.fm/

combined with relevance feedback. The system proposed by the authors was a content-based recommender using TreeQ method to quantize audio features into a vector value. They generate a tree that contains songs the user likes (good songs) and hates (bad songs). Based on the tree, a feature vector is made representing user preference as "good songs" and "bad songs". After the song features are represented in a vector, the authors applied cosine distance measure to calculate the similarity of songs. Even though the TreeQ method is effective for music information retrieval, it requires 100 or more training data to generate the tree. It is difficult to find a user who rates hundreds of songs. So, in order to build user profile using TreeQ structure, the authors adjusted user's preference model using relevance feedback. The experiment of the proposed system was conducted on 756 songs collected from Japanese online CD shop[7]. 12 subjects are participated to collect user preference, they are asked to rate the songs in the range of 1 and 5. The authors evaluated the proposed system and found precision of more than 40% for music information retrieval and it's improved when it is supported by relevance feedback to personalize user's preference. However, the proposed method (TreeQ) is slow, especially when the set of songs is large. This will be very time taking when it comes to the real world data. The user preference of songs is measured based on only genre.

Another content-based music recommendation system is proposed by Bogdanov *et al.* [73]. It generates recommendations based on semantic descriptors of the songs. The authors proposed two distance-based recommendation methods using weighted Pearson correlation distance and probabilistic model. The first approach, Semantic distance from the mean, models the user preference across individual tracks to a single point in the semantic descriptor space. The second approach, Semantic distance from all tracks, models user preference across all individual tracks. The third approach, Semantic Gaussian mixture, model the user preference as a probability density of preferences in the semantic space. The data for this experiment was collected by user's who are asked by the researcher to gather their preferred set of songs either in the form of audio or editorial metadata. Each of the users gathered a varied number of tracks ranging from 23 to 178. The low-level audio features of this songs were extracted using In-House Audio Analysis Tool[8]. The authors requested 12 subjects to evaluate the proposed system. The evaluation was made against two metadata-based and two audio content-based baselines. They claim that, based

[7] HMV Japan, contents in Japanese only (http://www.hmv.co.jp/)
[8] Music analysis, transformation and synthesis tool (http://mtg.upf.edu/technologies/essentia/)

on the response obtained from the participants, the proposed system gives a promising result. Dependency on editorial metadata of songs is one of the limitations of this work. Maintaining editorial metadata requires follow up, more time as well as expert participation.

3.3 Context-Based Approach

Reddy and Mascia [14] proposed a context-aware mobile music player that uses a context-sensitive music engine to drive recommendations suitable for the users and the location where they currently reside in. Generally, the system has four main components: userspace document, context engine, rating generator and music player. The userspace document contains a list of songs and associated contexts. The context engine collects data considering the location of the user, the time of operation, the velocity of the user, and environmental information such as traffic, weather, and sound modalities. The authors used mobile sensors to collect the contextual information such as GPS, RSS, clock, and microphones. Rating generator creates a playlist using data from user space document and current context. The music player serves as an interface between the user and the system. The context engine analysis user feedback to model user preference of songs for a particular context. Recommended list of songs is created based on the tags made by the users on the tracks, albums or artist in the user's library. The user preference is determined by the value of the tags related to the context parameters. The values either increase (e.g. if the tag is like) or decrease (e.g. if the tag is hate) importance of a recommended song for current user context. The drawbacks of this research were, first, it's not evaluated so it's hard to decide whether it performs well or not. The other problem is the manual tagging of the songs, which is labor-intensive and time taking.

Another music CAF system was proposed by S. Lee and C. Lee called Music for My Mood (M^3) [12]. M^3 has three layers, the interface layer, application layer and repository layer. The interface layer collects contextual data and presents the recommended songs to the user. The repository layer stores and manage user context data, listening history and music data. The application layer generates recommendation by inferring the user context. Generally, the proposed method assumes that if a user listened to certain songs in certain contextual situations, the same user will want to listen to similar songs in similar contextual situations. Then the user preference is determined from their listening experience. For the purpose of evaluation and testing, the authors used 14,373 records of listening history. The dataset contains a list of songs and their genres listened by certain users. The weather data is also collected from weather bureau that contains season, month, day of the week, atmospheric conditions and lowest, highest and average temperature information. The proposed approach was evaluated by comparing it with a case-based reasoning music recommender without contextual information. From the result obtained the proposed approach outperformed the recommender system without contextual information. However, the system can only classify songs that are slow or fast, this may not satisfy users with a preference other than such type of songs.

Kaminskas and Ricci [74] proposed a place of interest (POI) and emotional tag aware music recommender system. The authors used emotional tags that match both music and POIs. They assume that both music and place can affect mood so that the common property of this two is used as a base for generating a list of recommended songs. In order to match POIs with music, the authors used similarity metrics such as Cosine similarity, Dice similarity, Jaccard similarity that are thought to be applicable to tagged resources. The authors performed an experiment of the proposed system using 75 classical song tracks and movie soundtracks, and 50 POIs in the city of Bolzano and surrounding areas. The tagging was performed by 32 volunteer students and researchers of the Free University of Bolzano. The researchers evaluated the proposed system in two steps both offline and online. The offline evaluation was by comparing the similarity metrics offline. In online evaluation, subjects are asked to evaluate if the recommendation made for a particular POIs is satisfactory. The authors' claim that, the users were satisfied with the recommendation made, especially with the recommendation made by using Jaccard similarity measure. The main drawback of this work is recommending the same kind of songs for all users in a similar contextual situations, recommendation is not personalized.

Baltrunas *et al*. [30] Proposed Android-based mobile music recommendation system, InCarMusic. InCarMusic recommend music to the passengers of a car by inferring the ratings they give for songs. In the long term, it also adapts recommendation for the user who didn't provide any ratings. The authors considered different contextual factors that could influence music preference of the user. These contextual factors include driving style, road type, landscape, sleepiness, traffic conditions, mood, weather, natural phenomena. In order to generate personalized context-aware recommendations, the authors adapted and extended the Matrix Factorization approach by incorporating the contextual factors. For the purpose of experimenting and evaluation of the proposed approach, the authors used a total of 131 songs from MusicLoad[9]. 50 songs for relevance assessment of different contextual factors and 89 songs for the assessment of the impact of contextual conditions for particular tracks. The proposed approach is evaluated offline. From the result, the proposed system outperformed the matrix factorization method without contextual factors by 3% improvement. However, the system is dependent on rating of songs which requires a lot of user effort.

Martin *et al*. [15] proposed music recommender system that incorporates context-based music information that is extracted from playlist names into the recommendation process. The authors clustered the playlist into contextual clusters using k-Means clustering. The recommendations are generated using collaborative filtering by applying Jaccard similarity measure between the users. The authors created a dataset by extending the openly available nowplaying[10] dataset and Spotify. That contains information about 15,345 user who listened 1,878,457 unique tracks of 276,848 unique artists contained in 143,528 unique playlists. The evaluation of the proposed approach is performed by comparing three different recommendation systems: a pure CF without clustering, a top k-clustering recommender, and a recommender with CF and top 12-clusters. The authors claim that the contextual information extracted from playlist names has improved the music recommender reaching precision values 33% higher than traditional approaches. However, the music content information in this research is not considered at all.

Jiang and Yuan [17] proposed context-aware music recommender for smartphones, Smart-DJ. The system maps discretized contextual information (activity level, noise, time, social contact) with an audio music features (tempo, pitch, MFCC) through the Increment Regression Tree model. They collected three types of data from the user while they are listening to a song. These data include listening context, the audio features of currently playing song and rating given to the song as a feedback. Then these data are used to personalize the recommendation in addition to audio features. In order to implement the proposed system, the authors collected context data and a dataset of 876 songs from different music streaming websites. Evaluation of Smart-DJ is performed in two phases. First, it’s evaluated by 16 volunteer graduate and undergraduate students through a real-world experiment. The second evaluation is made by comparing the proposed recommender with a random recommender that randomly generates recommendations.

[9] http://www.musicload.de/
[10] http://dbis-nowplaying.uibk.ac.at/

The authors claim that the proposed system outperforms the state of the art methods. One of the limitations of the research is, it doesn't take the user mood into account.

3.4 Hybrid Approach

Andjelkovic et al. [75] proposed "MoodPlay" that combines content and mood-based filtering. It allows users to input names of artists to create their profile and they are positioned in a pre-computed latent mood space. Then recommendations are based on the position of the user on the mood space. They applied the Euclidean distance measure to analyze mood-based similarity between artists. The authors used randomly selected dataset of 4927 songs from Million Songs Dataset[11] and additional popular artist's songs from EchoNest Database[12]. They extracted mood-related information of each song from Rovi API[13] and the artists are linked to their profile on Last.fm. The evaluation of the system is made by 240 paid participants ages ranged from 18 to 65 of 43% female. The authors claimed that the participants generally liked the recommendation provided by the system. However, the recommender doesn't automatically change suggestion depending on user's current mood. User's mood is determined from the artists they enter into the system.

3.5 Recommendation System Using DS Theory

In fact, we didn't find any music recommendation system using DS theory. But there are attempts of using the DS theory in recommendation system for other domains. Sagdoldanova *et al.* [76] are among the researchers who used DS theory for medicine recommendation. The system they proposed recommends medicines based on symptoms of diseases provided by the user. The authors assigned a belief degree for medicines that are recommended for a headache, toothache, flu, and fever. Recommendation of medicine is generated by combining the beliefs according to the symptoms. The authors collected data from pharmacists and experts about frequently occurring diseases in Kazakhstan and recommended medicine that can be taken without physicians' prescription for each type of disease. The authors claim that the proposed system minimizes efforts of pharmacists and saves time. However, there is no evaluation information is provided for the proposed system.

---

[11] Million Songs Dataset: a freely-available audio features and metadata of popular music tracks (https://labrosa.ee.columbia.edu/millionsong/)
[12] EchoNest: a music intelligence and data platform (http://the.echonest.com/)
[13] Rovi API for entertainment information (http://prod-doc.rovicorp.com/mashery/index.php/Rovi-Data/)

Guo *et al*. [64] proposed a mobile-based e-commerce recommender that exploits multi-source information to analyze consumer's preference. The recommendation is generated by computing evidence weights using radial basis function network and DS theory to combine the multi-source information. For the purpose of experimentation, the authors collected 100 customers information from online shopping website, Taobao[14]. The category of collected information includes, location-service, social-platform, commodity-picture and the data from the store trading database. The authors evaluated the proposed method with the previous traditional recommenders. Based on the criteria of recommendation accuracy, coverage rate, simplicity, and recall rate, the proposed approach outperformed the traditional recommender.

3.6 Summary

As discussed in this chapter previous researchers strived to work on music recommendation system that eases effort and narrow choices to those that best satisfy users' needs. However, there are still gaps that require more studies. Even though, music mood perception is influenced by different cultural factors, there is no mood based music recommender, considering Ethiopic music into account. Music mood perception differs from person to person. However, some of the previous studies [69, 70] determine user preference based on others preference rather than their own perception. This is an open question, that requires additional problem-solving research [11]. There are research works that recommend songs based on context information [17, 30]. In order to provide a more accurate recommendation, the recommenders based on the CF approach require more time and data. Since CF approach needs user action on the songs (listening, rating etc.) to construct a user-item matrix, new songs or users suffer from cold start problem. Which means, new user or a song item may have few actions, therefore it will be difficult to relate the user or item with the others. On the other hand, as the data gets larger, only a few of them get the action of a user, again computing similarity of users or songs on a few common actions will be unrealistic. Recommendations based on CF approach also have a problem of popularity bias due to the songs that have many user actions. Popular songs with many user actions tend to be recommended more frequently while little-known songs remain obscure. The recommenders based on CBF approach usually give similar recommendations. Since most of user preference modeling is built in content similarity,

[14] http://www. taobao.com/

the recommender always generates only too similar or obvious songs. In general, previous works have contribute a lot for this research area. As it's indicated in the review all types of filtering approaches discussed above, have their own advantage and disadvantage. A lot of research communities in this domain suggest to hybridize the approaches in the mood based music recommendation system.

Chapter 4: Mood Based Hybrid Ethiopic Music Recommender

4.1 Overview

Mood Based Hybrid Ethiopic Music Recommender (MBHEMR) is a mobile recommendation model. It recommends music considering the current mood of the smartphone holder. The recommendation model is based on mood-based classification of music taking into account Ethiopian culture and music listening habit.

The main tasks in MBHEMR are categorized into three: 1) Classifying Ethiopic music into different mood types by extracting audio music features. This is done using machine learning techniques and it’s detailed in Section 4.2.1 and 4.2.2; 2) User context managiment including usre mood detection from contextual finformation as detailed in Section 4.2.3, and 3) Recommending a list of music that fits the current mood of the user as detailed in Section 4.2.4.

4.2 Architecture of MBHEMR

The proposed Music Recommender has five main components. These include:

- Feature Extraction and Selection component
- Music Mood Recognition Model (MMR model)
- Music Mood Annotator
- Context Manager and
- Recommender

Figure 14 depicts the general architecture of the MBHEMR having the high-level components that form the entire system.

4.3 MMR Model

4.3.1 Moods Conveyed in Ethiopic Music

As discussed in Section 2.3.1, one of the preliminary tasks of music mood recognition process is defining a possible set of mood types conveyed in music. However, to define moods in Ethiopic music, there is no taxonomy of music mood in Ethiopian context. Therefore, we chose the Tellegen, Watson, and Clark categorical music mood model from currently available music models, then customized it by conducting a survey, which participates psychologists and music experts as detailed in Section 5.1.1. The Tellegen, Watson, and Clark music mood model is chosen because of the familiarity of the affect words in the model for the survey participants. The limited number of the affect words is easy to control and manage. Moreover, the words are used in everyday language of the public [71]. After making modification on the music mood model, 5 primary and 25 secondary mood adjectives are found. These include: HPA (Active, Elated, Enthusiastic, Excited, Peppy, and Strong), LNA (At rest, Calm, Placid, and Relaxed), P (Content, Happy, Kindly, Pleased, Satisfied, and Warmhearted), SE (Aroused, Astonished, and Surprised) and UnP (Blue, Grouchy, Lonely, Sad, Sorry, and Unhappy).

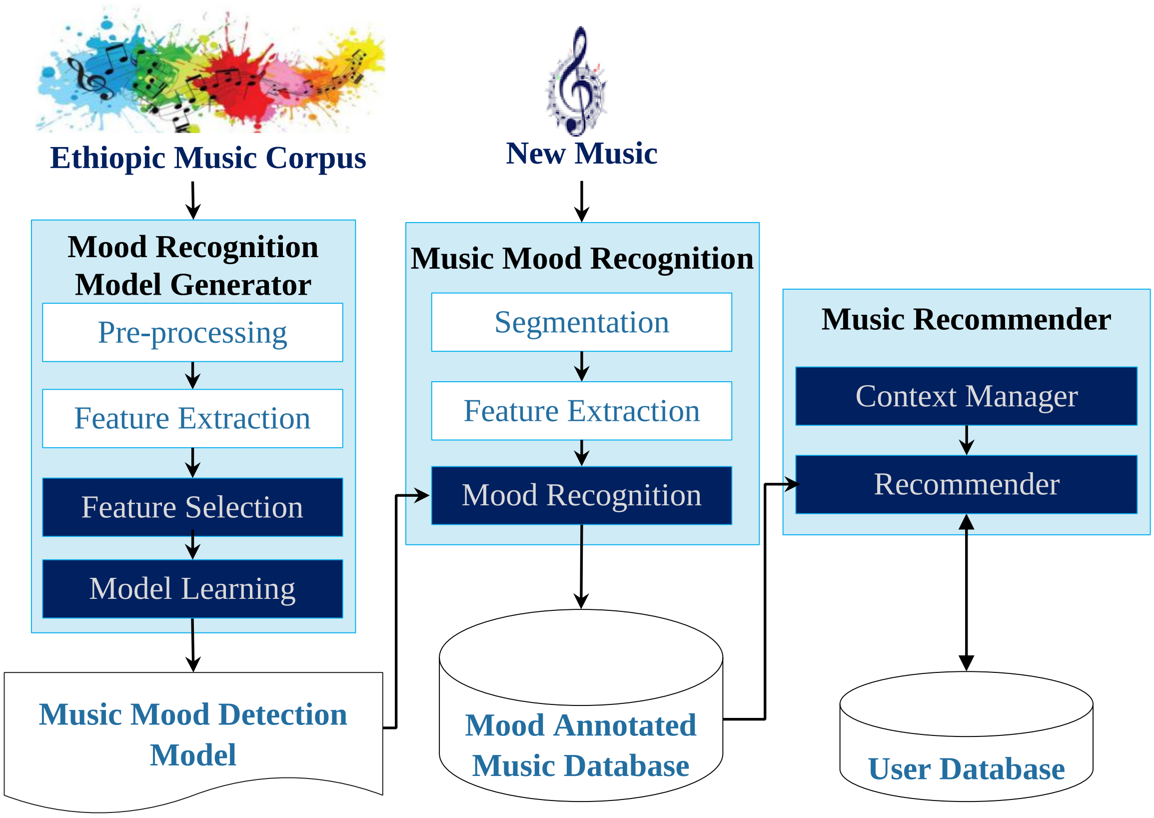


***Figure 14**: High-level Architecture for MBHEMR*

4.3.2 Pre-Processing

Audio pre-processing of songs is performed to make the song ready for feature extraction. The pre-processing stage converts a stereotype songs to monotype and different audio format to .wav format. It's common that songs are recorded both in stereotype (e.g. Kuku Sebsibe's track recording "Libee segga" whose wav format is depicted in Figure 15) or monotype (e.g. Tsedeniya G/Markos's track recording "Mirchaye" whose wav format is depicted in Figure 16). Since feature extraction is not supported from the stereotype by the technique we use, the basic audio processing is started by checking the audio format and the number of channels used to record the song.

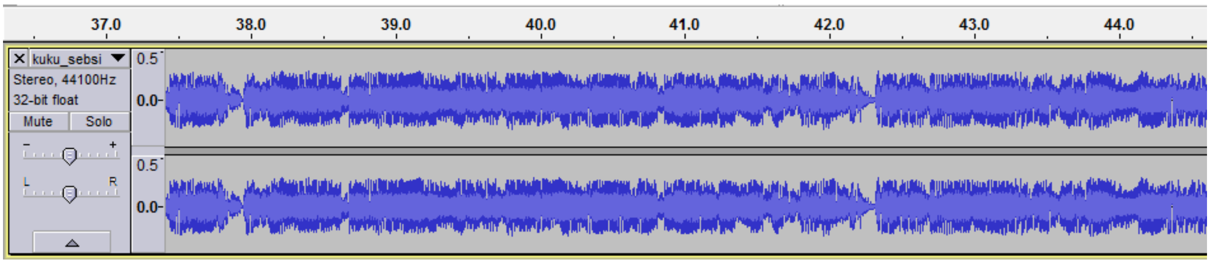


Figure 15*: Unprocessed stereotype song segment*

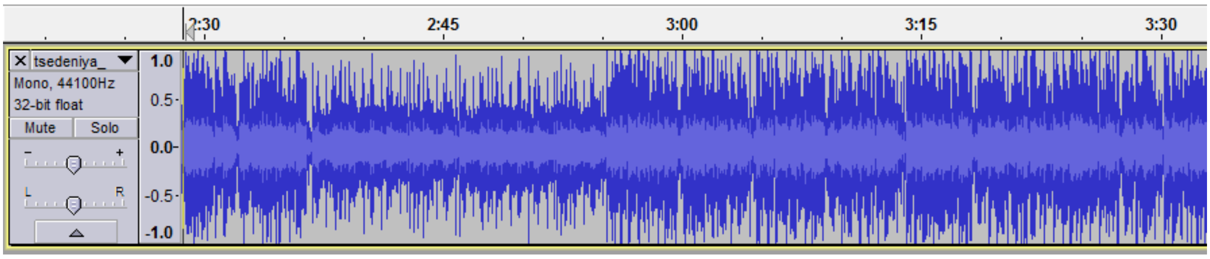


Figure 16*: Monotype song segment*

If the songs are in a format other than the .wav, they are automatically converted to .wav form. If more than one channel is used to record the song (stereotype song), it's automatically converted to one channel (monotype song).

4.3.3 Feature Extraction

The feature extractor generally accepts audio music from the pre-processor as input and provides audio feature information as an output. A total of 42 dynamics, rhythmic, timberal and spectral music features are extracted (Annex C) by decomposing the audio song along frequencies in each frame then analyzing the spectrum and correlation of each frame. Among the 42 features 11 are selected to construct the MMR model as discussed in Section 4.2.1.4. The process of feature extraction is slightly different for each distinctive feature types but most of them have a common step which is depicted in Figure 17. We used the methods of Lartillot *et al*. [52] to extract the features for this research.

Frame Decomposition*:* in the feature extractor component, the first step after the pre-processed audio song input is, decomposing the audio into different frames length. Analyzing the whole temporal signal provides only overall description of the average feature value of the song. Since music event changes throughout the track, the song should be decomposed into frames. This helps to consider the dynamic evolution of affective music features distributed in the music track.

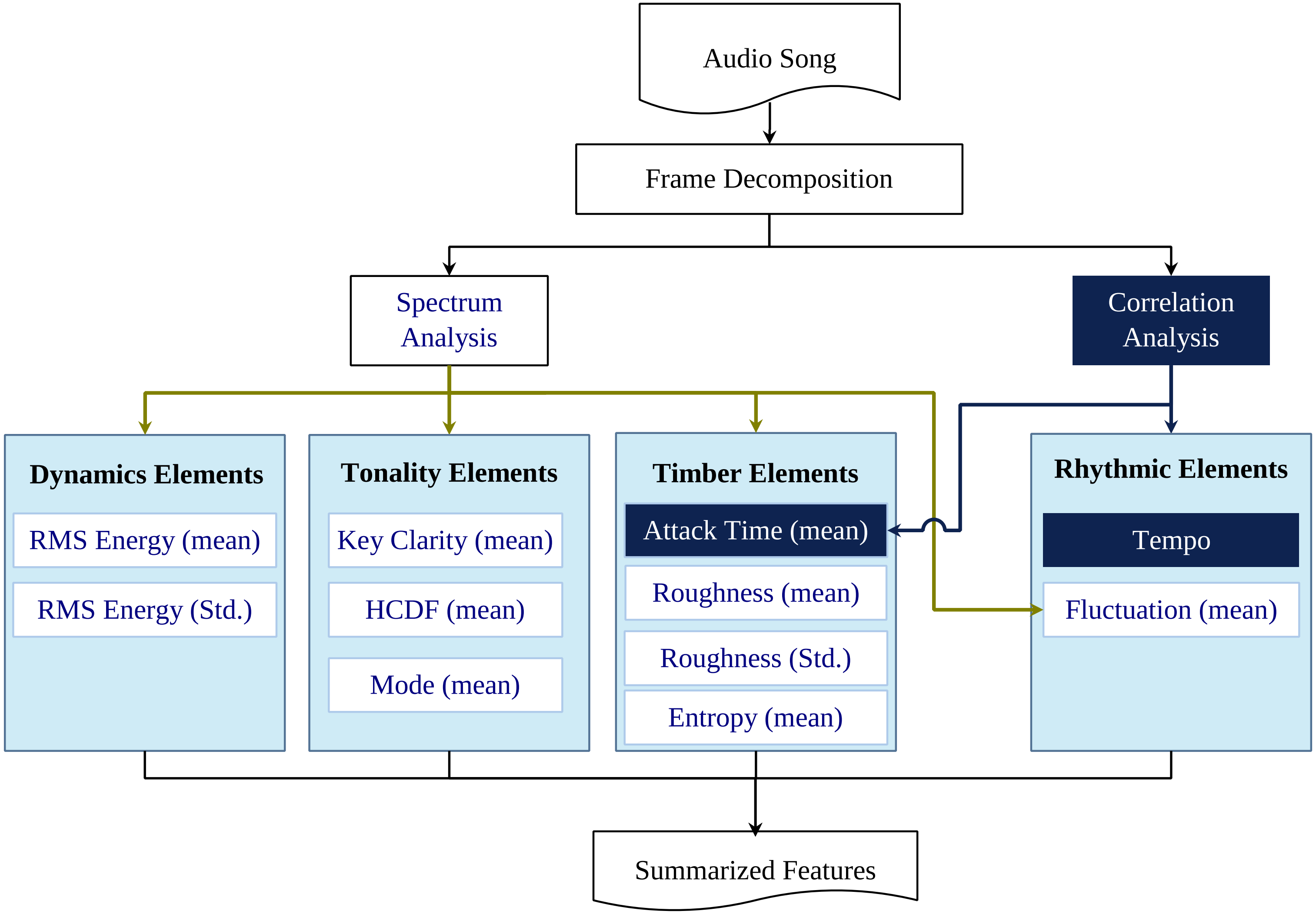


***Figure 17**: Steps of the feature extraction processes*

Spectrum Analysis*:* the Fast Fourier Transform function used to decompose the audio song along frequencies in each frame. Spectrum analysis of these frames gives music features, such as mean of fluctuation, spectral spread, spectral centroid, entropy of spectrum, roughness, key clarity, mode, Harmonic Change Detection Function (HCDF), Root Mean Square energy (RMS energy) and others.

Correlation Analysis: analyzing correlations between events in the audio signal provides periodicity information. Tempo and mean of attack time are among the periodicity information that can be found by correlation analysis of frame decomposed audio song.

4.3.4 Feature Selection

In the previous studies [72, 73, 77], researchers proposed different types of features in a song to train automatic music mood classifying model depending on the context of their research. Here we proposed the music features that can affect user perception towards music mood in case of Ethiopian music listening habit.

Among the 42 features we extracted (Section 4.2.1.3, Annex C), 11 of the best features are selected that are thought to provide high performance MMR model. These features include RMS Energy (Std. and mean), fluctuation (mean), attack time (mean), the entropy of spectrum (mean), roughness (Std. and mean), mode (mean), key clarity (mean), HCDF (mean), and tempo.

The stepwise regression method of forward selection approach applied to select the features with minimum error rate. Table 2 shows, which features contribute to discriminate one class from the other for each pair of classes. Table in Annex D also presents, properties of the feature values corresponding to each type of music mood. Therefore, construction of the MMR model is based on the data presented in Table 2 and Annex D.

***Table 2**: Feature to Discriminating one type of music mood from the other*

HPA vs. LNA	RMS Energy, Fluctuation, Entropy of Spectrum, Key Clarity, Mode
HPA vs. P	RMS Energy, Fluctuation, Attack Time, Entropy of Spectrum, Roughness, Key Clarity, HCDF, Tempo
HPA vs. SE	RMS Energy, Entropy of Spectrum, Roughness, HCDF
HPA vs. UnP	RMS Energy, Fluctuation, Attack Time, Entropy of Spectrum, Roughness, Key Clarity
LNA vs. P	RMS Energy, Fluctuation, Attack Time, Entropy of Spectrum, Key Clarity, HCDF, Tempo
LNA vs. SE	RMS Energy, Entropy of Spectrum, Roughness, Mode, HCDF, Tempo
LNA vs. UnP	RMS Energy, Attack Time, Entropy of Spectrum, Roughness, Roughness, Mode
P vs. SE	RMS Energy, Fluctuation, Entropy of Spectrum, Roughness, HCDF
P vs. UnP	RMS Energy, Fluctuation, Attack Time, Entropy of Spectrum, Key Clarity
SE vs. UnP	RMS Energy, Entropy of Spectrum

4.3.5 Model Learning

Once the music features are extracted and selected, supervised machine learning technique, linear SVM, is applied to train MMR model. The model learning component accepts manually labeled music mood and features that are extracted using the feature extractor then generates the MMR model.

SVM is a vector based algorithm, in order to perform the music mood classification, the boundaries between classes of mood in a feature space are identified from the music dataset feed to it. SVM is also a binary classification algorithm, therefore we converted our classification problem into binary using one vs one (all pairing) approach. The overall model learning process is graphically shown in Figure 18.

Selected Music Features (Dataset): the input for model learning module is a dataset. The dataset generally contains 11 features/predictors (Section 4.2.2.3) and 5 responses (HPA, LNA, P, SE and UnP). Different type of features used for each pair of classification model, accordingly as presented in Table 2 (Algorithm 1 line 4).

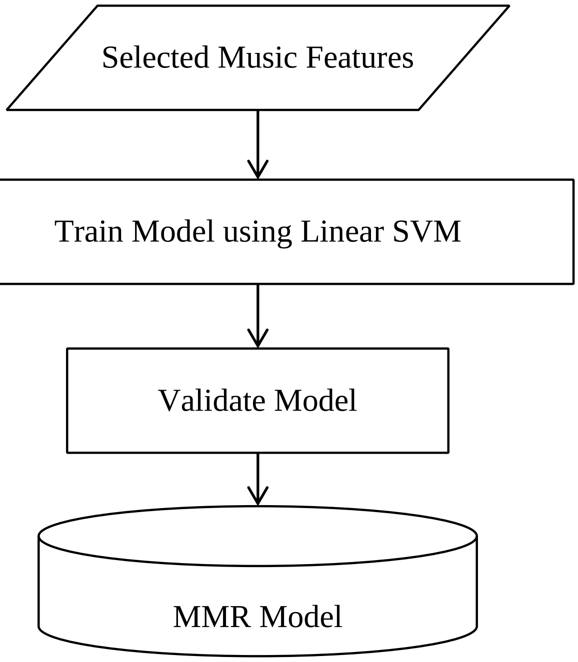


***Figure 18**: MMR model learning process*

***Algorithm 1**: MMR model learning*

```
Input: dataset
Output: MMRmodel
Begin
Pair[] = svm_ovo_mapping(HPA, LNA, P, SE, UnP)
i = 0
do{
      data[] = dataset.Pair[i]
      MMRmodel = append.train_SVM(
          kernel = linear
          predictor = data[predictors]
          response = data[mood]
          standardize = true
          mapping = One_vs_One
          classes[] = {Pair [i]}
      )
     i++
}while(i < 10)
End
```

Train recognition model using SVM: the model learner selects support vectors using linear SVM from the given dataset to come up with hyper-planes or optimal boundaries boundary line between each pair of classes using the formulation in Equation 12 (Algorithm 1 line 6-11).

$$f(X) = \left(\frac{Z}{s}\right)' \beta + b \qquad \textbf{\textit{(12)}}$$

Where Z is the standardized value of the feature data (X). s, b (Table 3) and β (Annex D) are kernel scale, bias and beta coefficient respectively.

In order to improve performance of the MMR model, the feature data is normalized using the standardization method of Equation 13.

$$Z = \frac{X - \mu}{\sigma} \qquad \textbf{\textit{(13)}}$$

Validate model: to determine whether the generated model performs good or not, the cross validation method is applied, with a k value of 10. The model validation module returns accuracy of the result in percent. This is also used to compare the proposed algorithm with other algorithms for evaluation purpose in Section 5.1.2.

MMR Model: is the output of the model learner. It includes the music mood prediction function. That referred by the Music Mood Annotator component in order to predict the mood of new music.

Table 3*: Bias and Kernel Scale of the hyper-planes for each pair of classes*

	Bias (*b*)	**Kernel Scale (*s*)**
HPA vs. LNA	-0.2540	1.1449
HPA vs. P	-0.7355	1.7281
HPA vs. SE	-0.7530	0.7648
HPA vs. UnP	-0.1620	1.1807
LNA vs. P	-0.4858	1.7120
LNA vs. SE	-0.6420	1.4068
LNA vs. UnP	-0.0048	0.8872
P vs. SE	-0.1446	1.2146
P vs. UnP	0.5961	1.1026
SE vs. UnP	0.7323	0.5018

4.4 Music Mood Annotator

The music mood annotator component predicts the mood of a new music referring to the MMR Model. It takes an excerpt of the new song by segmenting the audio and extracts selected features to determine the mood of the song by calling music mood prediction function of the MMR model (Algorithm 2 line 1).

***Algorithm 2**: Music mood Annotator*

```
Input: audio_song,
       segment_start
Variable: feature[10]
Output: music_mood
Begin
 1. audio_input = audioRead(audio_song)
 2. wav_file = audioConvert(audio_input, 'audio_wav.wav')
 3. songExcerpt  = audioSegment(wav_file, segment_start){
 4.  segment_end = segment_start + 30
 5.  return trimAudio(wav_file, segment_start, segment_end)
 6. }
 7. feature [0] = mean(getRMSEnegy(songExcerpt))
 8. feature [1] = std(getRMSEnegy(songExcerpt))
 9. feature [2] = mean(getFluctuation(songExcerpt))
10. feature [3] = mean(getAttackTime(songExcerpt))
11. feature [4] = mean(getEntropySpectrum(songExcerpt))
12. feature [5] = mean(getRoughness(songExcerpt))
13. feature [6] = std(getRoughness(songExcerpt))
14. feature[7] = mean(getKeyClarity(songExcerpt))
15. feature [8] = mean(getMode(songExcerpt))
16. feature [9] = mean(getHCDF(songExcerpt))
17. feature [10] = getTempo(songExcerpt)
18. music_mood = MMRModel.musicMoodPredFun(feature[])
End
```

Segmentation: the music mood recognition algorithm begins with reading and converting the audio song into .wav format (Algorithm 2 line 1). Then segmentation of the audio begins starting from the segmentation starting position given in second (Algorithm 2 line 2-6). The starting position by default is zero or else other starting position can be specified, if the audio song contains unnecessary content (e.g. advertisements, long silence).

Selected Feature Extraction: unlike in the MMR model construction, in this component, only eleven selected features are extracted from the audio excerpt for the music mood perdition (Algorithm 2 line 7-17). Table 4 presents an instance of selected audio feature values from tizita.wav file.

Table 4*: Selected audio features of tizita.wav*

RMS Energy (Mean)	0.114917	Roughness (Std)	282.855747
RMS Energy (Std)	0.03692	Key Clarity (Mean)	6.827586
Fluctuation (Mean)	183.985401	Mode (Mean)	-0.050818
Attack Time (Mean)	0.019299	HCDF (Mean)	0.376289
Entropy of Spectrum (Mean)	0.668203	Tempo	93.749099
Roughness (Mean)	291.526368		

Music Mood Annotation: to determine the mood of a given song the music mood prediction function of the MMR model is called by passing the selected features. For instance, mood of the song (tizita.wav), whose features are presented in Table 5, can be determined by using appropriate features for each pair of music mood class as follows:

For HPA vs LNA X= [RMS Energy (Mean): 0.114917, RMS Energy (Std): 0.03692, Fluctuation (Mean): 183.985401, Entropy of Spectrum (Mean): 0.668203, Key Clarity (Mean): 6.827586, Mode (Mean): -0.050818]. Based on the values given in Table 3, standardizing X gives its normalized value of Z = [RMS Energy (Mean): -0.2137, RMS Energy (Std): -0.6690, Fluctuation (Mean): -0.3593, Entropy of Spectrum (Mean): -1.2930, Key Clarity (Mean): 0.1837, Mode (Mean): -0.5458].

Then referring the bias and kernel scale values in Table 4 and the beta coefficient values in Table 3 we get a classification score of $f(X)$ = 0.9332. In this case, if a song has a classification score $f(X) < 0$ in the negative region of the hyper plane, its HPA type of song, for this particular pair of class. Otherwise if $f(X) > 0$ the song is UnP type. Therefore, the song tizita.wav is UnP type in this particular pair (HPA vs LNA) so that a vote is counted for UnP mood type.

Computing classification score and predicting class of a song can be done using the same method for the other pairs of music mood types. Finally mood of the song is decided by combing the votes using weighted majority vote approach. Vote from each pair of classifier is multiplied by their classification confidence. Then the song is annotated with a class with the largest score.

4.5 Context Manager

The context manager identifies and provides the user context. This component predicts user music mood taste based on contextual data of a user, sensor reading, and music player list. The user provides static context data in the form of demographic information and dynamic mood signals in the form of textual query sentences. Heart-rate data is collected automatically when the user measures and store his pulse using the heart-rate sensor or can be provided by the user using the music exploring interface. The music player provides

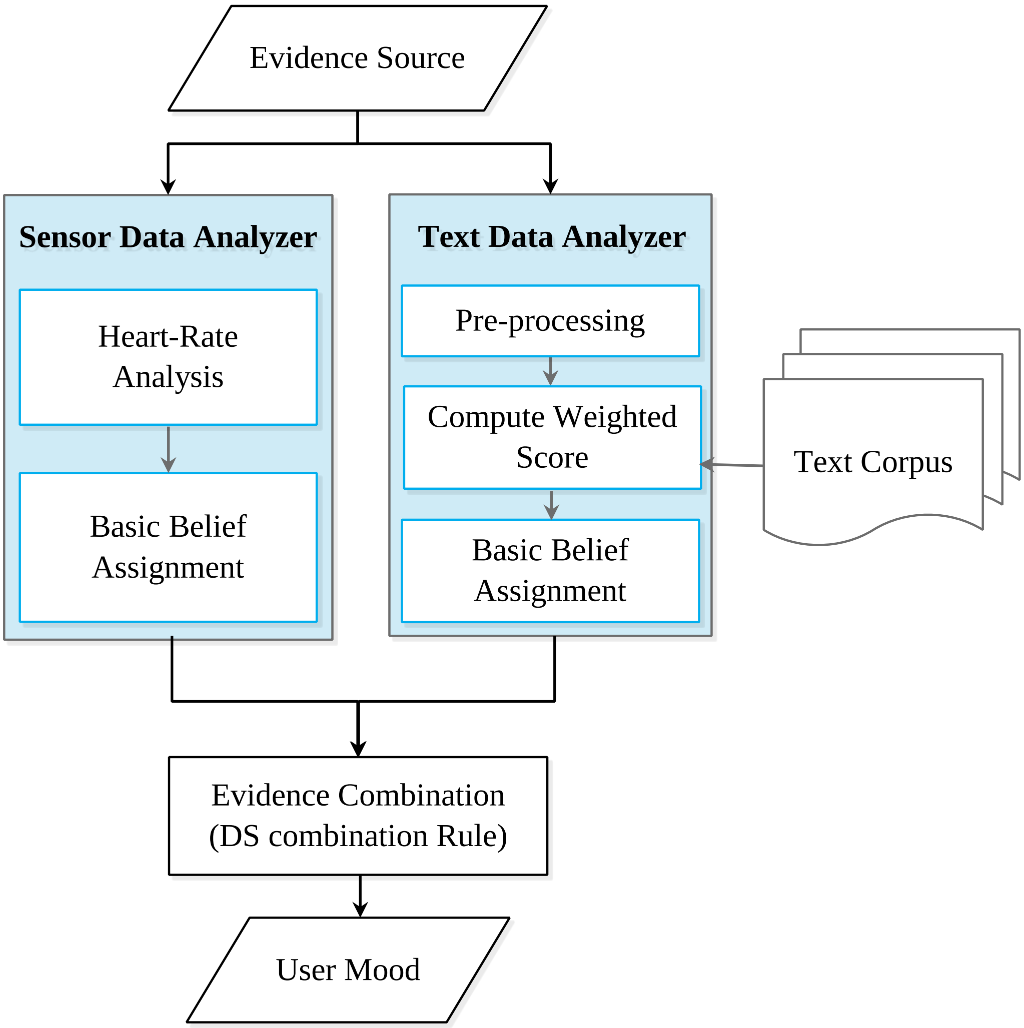


***Figure 19**: User mood detection process*

preference of the user, listening history both current and favorite, frequent play list etc.

User mood can be expressed through different modalities such as text, biometrics/physiological changes, speech audio, facial features, and body gestures. In order to detect the user mood, we proposed to combine biometrics (heart-rate) and text expression modalities.

The mood expressing modalities, heart-rate, and text, are from different sources (sensor, user or music player) and have a distinct type. Some information (heart-rate or text) might be missing or unavailable from one of these sources. The heart-rate information, we are using is experts view (its generic) that may have exceptions.

Recommendation based on such data with uncertainty issues might lead to wrong decision. Therefore, these issues must be considered during user mood detection. So that, we proposed mood detection method based on the DS theory. That enables handling these uncertainties and incorporation of shared beliefs from the different sources. Figure 19 depicts the user mood detection method using the DS theory.

In DS theory, problem formalization begins with identifying all possible conclusions to be drawn i.e. determining possible user mood from heart-rate and text. The user mood towards music can be classified into HPA, LNA, P, SE, and UnP. This is formalized as a set $H = \{HPA, LNA, SE, P, UnP\}$.

The frame of discernment can be derived from all possible subset of the universal set H and is represented in power set

$$2^H = \{\{\phi\}, \{HPA\}, \{LNA\}, \{P\}, \{SE\}, \{UnP\}, \{HPA, LNA\}, \{HPA, P\}, \{HPA, SE\}, \{HPA, UnP\}, \{LNA, UnP\}, \{LNA, P\}, \{LNA, SE\}, \{P, SE\}, \{P, UnP\}, \{SE, UnP\}, \{SE, UnP\}, \{HPA, LNA, P\}, \{HPA, LNA, SE\}, \{HPA, LNA, UnP\}, \{HPA, P, SE\}, \{HPA, P, UnP\}, \{HPA, SE, UnP\}, \{LNA, P, SE\}, \{LNA, P, UnP\}, \{LNA, SE, UnP\}, \{P, SE, UnP\}, \{LNA, P, SE, UnP\}, \{HPA, LNA, P, UnP\}, \{HPA, LNA, P, SE\}, \{HPA, LNA, SE, UnP\}, \{HPA, P, SE, UnP\}, \{HPA, LNA, P, SE, UnP\}\}.$$

The problem formalization is followed by evidence collection, analyzing the evidence, basic belief assignment and combination of shared beliefs, respectively. All of these steps are discussed in detail as follows.

4.5.1 Sources of Evidence

The main evidences considered in this work and support beliefs held on current user mood are heart-rate, text collected from the user smartphone and music listening history.

Sensor Data (heart-rate): the relationship between heart-rate and user mood are studied by many researchers [53, 54, 55, 56, 57]. In this work, we adopted the work of Wibawa et al. [54] and Valderas et al. [57] as depicted in Figure 20.

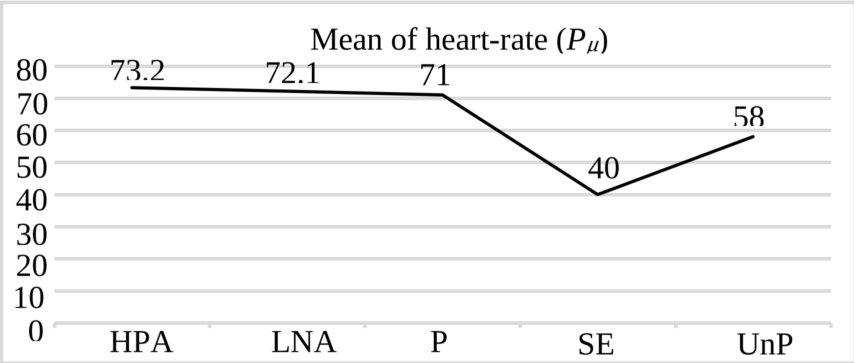


***Figure 20**: Association of Mood and heart-rate in bpm*

Text Data: text data is another source of evidence which is a music query posed by the user that bear the mood of the user. The association between text and user mood is performed using a model, which is built by applying the notion of TFIDF. The model is constructed on annotated corpus of English mood sentences that reflects one or more of user mood. It provides weighted score of a sentence related to a specific type of mood.

4.5.2 Belief from Sensor Data

Heart-rate sensor sense heart-rate data. A belief mass is assigned to all possible set of mood by analyzing and normalizing the heart-rate value. Algorithm 3 shows how the belief assignment is working, and it's discussed as follows.

Heart-rate Analyses: the heart-rate analyzer retrieves mean of most recent heart-rate from the user smartphone. Then the value of this heart-rate is compared with each mean of pulses presented in Figure 20 to find the closest type of mood. This is computed by calculating the difference between each mean of pulses and the new pulse (Algorithm 3 line 1-6). That indicates, if the difference between the new heart-rate and a mean of pulse of a particular mood type is minimum, the current user mood is closer to that particular type of mood. For instance, mean of the heart-rate recently scanned by the user is P_n = 74bpm, then this is computed as follows:

- Difference between P_μ of HPA and P_n (difHPA) = |P_μ of HPA-P_n| = |73.24-74|=0.76
- Difference between P_μ of LNA and P_n (difLNA) = |P_μ of LNA-P_n|=|72.13-74|=1.87
- Difference between P_μ of P and P_n (difP) = |P_μ of P - P_n| = |71 - 74| = 3

- Difference between P_μ of SE and P_n (difSE) = $|P_\mu$ of SE - $P_n|$ = |40 - 74| = 34
- Difference between P_μ of UnP and P_n (difUnP) = $|P_\mu$ of UnP - $P_n|$ = |58 - 74| = 16

Where:

- P_μ is mean of heart-rate
- P_n is mean of most recent user heart-rate

From the result obtained difHPA = 0.76 is the lowest value with respect to the others. This shows that current user mood is closer to HPA. These values are then normalized and used later in the BBA function.

***Algorithm 3**: Heart-rate analyses and basic belief assignment*

```
Input: heart_rate        //mean of recent user's pulse
Output: m_sensor[]       //mass of belief (degree of belief)
Begin
RevDisHPA = 1-abs((73.24-heart_rate)/(P_max-P_min))
RevDisP = 1-abs((71-heart_rate)/(P_max-P_min))
RevDisSE = 1-abs((40-heart_rate)/(P_max-P_min))
RevDisUnP = 1-abs((58-heart_rate)/(P_max-P_min))
RevDisLNA = 1-abs((72.13-heart_rate)/(P_max-P_min))
RevDisHLPSU = max(RevDisHPA, RevDisLNA, RevDisP,
RevDisSE, RevDisUnP)
totalRevDis = RevDisHPA + RevDisLNA + RevDisP +
RevDisSE + RevDisUnP + RevDisHLPSU
beliefDegree(RevDisHPA, RevDisLNA, RevDisP, RevDisSE,
RevDisUnP, RevDisHLPSU, totalRevDis){
  m[HPA] = RevDisHPA / totalRevDis
  m[LNA] = RevDisLNA / totalRevDis
  m[P] = RevDisP / totalRevDis
  m[SE] = RevDisSE / totalRevDis
  m[UnP] = RevDisUnP / totalRevDis
  m[HLPSU] = RevDisHLPSU / totalRevDis
  m_sensor[] = m[HPA, LNA, P, SE, UnP, HLPSU]
  return m_sensor[]
}
End
```

In order to assign a belief mass (degree of belief) for elements of the frame of discernment, the values obtained in heart-rate analyses is normalized in the range between 0 and 1 so that it will be tuned with the belief assignment function (Equation 4). Equation 14 summarizes the heart-rate analyses and normalization.

$$dif = \frac{|P_\mu - P_n|}{P_{max} - P_{min}} \quad \textit{(14)}$$

Where:

- P_μ mean of pulses corresponding to each mood type
- P_n mean of most recent heart-rate of the user
- P_{max} and P_{min} are maximum and minimum records of user's heart-rate respectively.

Partitioning of belief degree is made according to the normalized distance (dif vale) of the mood categories. This is done by subtracting the normalized distance from 1 to reverse the value, so that the lower distance a category have the higher belief degree will be assigned.

Basic Belief Assignment: after partitioning of belief degree, a belief mass (m) is assigned to each possible subset of H (elements of the power set 2^H).

Since sum of the belief mass of elements in the power set 2^H should be 1 (Equation 6), the reversed distance values of the power set elements such as HPA, LNA, P, SE, UnP and H are divide by sum of the reversed distances of all elements of 2^H (Algorithm 3 line 10-16). These are assigned as a belief mass for corresponding HPA, LNA, P, SE, UnP and H according to their distance and the rest of the power set elements will a have 0 belief mass.

Continuing on the above example, assume P_{max} = 80 and P_{min} = 40:

RevDisHPA = 1 - (0.76 / 80 – 40) = 0.981

RevDisLNA = 1 - (1.87 / 80 - 40) = 0.953

RevDisP = 1 - (3 / 80 - 40) = 0.925

RevDisSE = 1 - (34 / 80 - 40) = 0.15

RevDisUnP = 1- (16 / 80 - 40) = 0.6

TotalRevDis = 0.981 + 0.953 + 0.925 + 0.15 + 0.6 = 3.609

Then the mass of belief for each element will be as follows:

m[HPA] = 0.981 / 3.609 = 0.272

m[LNA] = 0.953 / 3.609 = 0.264

m[P] = 0.925 / 3.609 = 0.256

m[SE] = 0.15 / 3.609 = 0.042

m[UnP] = 0.6 / 3.609 = 0.166

m[H] = 0.15 / 3.609 = 0.042

m[2^H| HPA, LNA, P, SE, UnP, H] = 0

***Algorithm** 4: Text data analyses and basic belief assignment*

```
Input: sentence
Output: m_sentence[]    // mass of belief (degree of belief)
Begin
tokens[] = tokenizer(sentence)
filtered_tokens = word for word in tokens if not word
in stopWords
dictionary = dictionary(filtered_tokens)
corpus = bagOfWord(dictionary)
tfidf(tokens[], sentence, corpus){
  tf =  term_frequency [word for word in tokens[],
  sentence)
  idf = idf (word for word in tokens[], corpus)
  tfidfScore = tf*idf
  return tfidfScore
}
beliefDegree(sentence, corpus){
  weight_hpa = tfidfScore (sentence, corpus_hpa)
  weight_lna = tfidfScore (sentence, corpus_lna)
  weight_p =  tfidfScore (sentence, corpus_p)
  weight_se = tfidfScore (sentence, corpus_se)
  weight_unp = tfidfScore (sentence, corpus_unp)
  weight_h = tfidfScore (sentence, corpus)
  total_weight = weight_hpa + weight_lna + weight_p +
  weight_se + weight_unp + weight_unp + weight_h
  m[HPA] = weight_hpa / total_weight
  m[LNA] = weight_lna / total_weight
  m[P] = weight_p / total_weight
  m[SE] = weight_se / total_weight
  m[UnP] = weight_unp / total_weight
  m[H] = weight_h / total_weight
  m_sentence[] = m[HPA, LNA, P, SE, UnP, H]
```

```
    return m_sentence[]
}
End
```

4.5.3 Belief from Text Data

This component also shares its belief but the source of evidence here is contextual text data that can reveal user mood from their context.

Generally, generating a belief degree from text involves pre-processing the input text, computing weighted score by referring the text-corpus, normalizing the weighted score and assigning the basic belief for elements of the frame of discernment. This is illustrated in Algorithm 4 and detailed as follows.

Pre-processing: the text pre-processing techniques that are applied to the sentences include: 1) Breaking up the sentence into words or tokenizing the sentence; 2) Removal of non-informative words/stop-words (e.g. 'the', 'a', 'to' etc.) and unnecessary characters such as '-', '…' etc.; 3) Creating a dictionary from the sentences to give every word a unique id; 4) Corpus preparation or creating bag of words (vector space model). The corpus / bag of words represents the sentences as a bag of its words, in which each unique word will have a corresponding number that represents it's occurrence in the sentence.

Computing Weighted Score: we used the notion of TFIDF to analyze and detect user mood expressed in a sentence (Algorithm 4 line 5-6). This method was first proposed by Ku and Sun [78], in their method, a sentence can express always a single type of mood and it's treated as a document. However, here we modified it by computing a probability of expressed mood in a sentence for each mood type. We used Gensim python library to compute weighted score according to the concept in Equation 15-18.

$$TF(t,s) = \frac{freq(t,s)}{\sum_i^n freq(t_i,s)} \qquad \textbf{\textit{(15)}}$$

$$IDF\ (t) = log(\frac{N}{N_t}) \qquad \textbf{\textit{(16)}}$$

$$TFIDF\ (t,s) = TF\ (t,s) * IDF(t) \qquad \textbf{\textit{(17)}}$$

$$W(s) = \sum_i^n TFIDF(t_i,s) \qquad \textbf{\textit{(18)}}$$

Where:

- *freq(t,s)* is the frequency of term t in a sentence s (in Equation 15).
- *TF(t,s)* the proportion of count of the term t in sentence s (Equation 15).
- *n* number of distinct terms in sentence s (in Equation 15 and Equation 18).
- N is every distinct number of sentences (in Equation 16).
- N_t frequency of sentences in which the term t is present (in Equation 16).
- W(s) (Equation 18) is the cumulative TFIDF (Equation 17) weight of terms in sentence s.

The weighted score is computed for each term in a sentence related to all individual classes of mood (HPA, LNA, P, SE, and UnP). Based on these weights, a basic belief assignment is made for each possible subsets of the power set 2^H using the following normalization formula (Equation 19) which is derived from Equation 6 (DS theory rule).

$$m(A) = \frac{w(s)_A}{\sum_i^l (W(s)_i)} \qquad \textbf{\textit{(19)}}$$

Where:

- $w(s)_A$ is the weight of the sentence related with subset $A \in 2^H$.
- $w(s)_i$ is the weight of the sentence related to the i^{th} element of the frame of discernment (2^H).
- l is a total number of focal elements (elements of the frame of discernment that have a non-zero mass of belief).

List of elements of the frame of discernment, such as *HPA, LNA, SE, P, UnP,* and *H* are assigned the belief degree as shown in Algorithm 4 line 19-26 using the concept in Equation 19. The rest of elements of the frame of discernment are assumed having zero degree of belief.

4.5.4 Evidence Combination

Combination of the basic beliefs is the last mass of beliefs used to decide the user mood. The basic beliefs are combined using the Dempster-Shafer rule of combination (Equation 8) as indicated in Algorithm 5.

The evidence combination module accepts the mass of beliefs from the heart-rate analyzer (Section 4.2.4.2) and the text data analyzer (Section 4.2.4.3). Then these beliefs are combined to predict user mood.

***Algorithm 5**: Combination of belief degrees*

```
Input: m_sensor[],
       m_sentence[]
Output: user_mood
Begin
 1. if (m_sensor[] ≠ ϕ & m_sentence[] == ϕ){
 2.   m = m_sensor[]
 3. }
 4. if (m_sensor[] == ϕ & m_sentence[] ≠ ϕ){
 5.   m = m_sentence[]
 6. }
 7. if (m_sensor[] ≠  ϕ & m_sentence[] ≠ ϕ){
 8.   m = DSCombination(m_sensor[], m_sentence[])
 9. }
10. m = m.pignistic()
11. user_mood = index(max(m[HPA, LNA, P, SE, UnP]))
End
```

The shared beliefs from the two analyzers are directly combined while conflicting beliefs that results with *'ϕ'* when they are combined are normalized by computing amount of conflict using Equation 9 (Algorithm 5 line 1-9).

Once the mass of beliefs combined, pignistic probability is computed for each elements of the universal set H from the combined / joint mass. Then user mood is determined from the value of the pignistic probability that provide/assign the maximum possible belief mass for HPA, LNA, P, SE and UnP. Type of mood that has a value of the maximum degree of beliefs is considered as the current user mood (Algorithm 5 line 10 and 11). The predicted user mood is then provided to the recommender component to generate a suitable list of songs.

4.6 Recommender

The recommender generates a suitable list of music for the user based on the current user context detected by the context manager component. The proposed approach combines content-based recommendation of songs with individual's music mood perception, experts view, and context. Figure 21 depicts the overall processes of the recommender component.

4.6.1 Retrieving Mood Based Songs

The first step in recommendation process is getting user mood from the context manager.

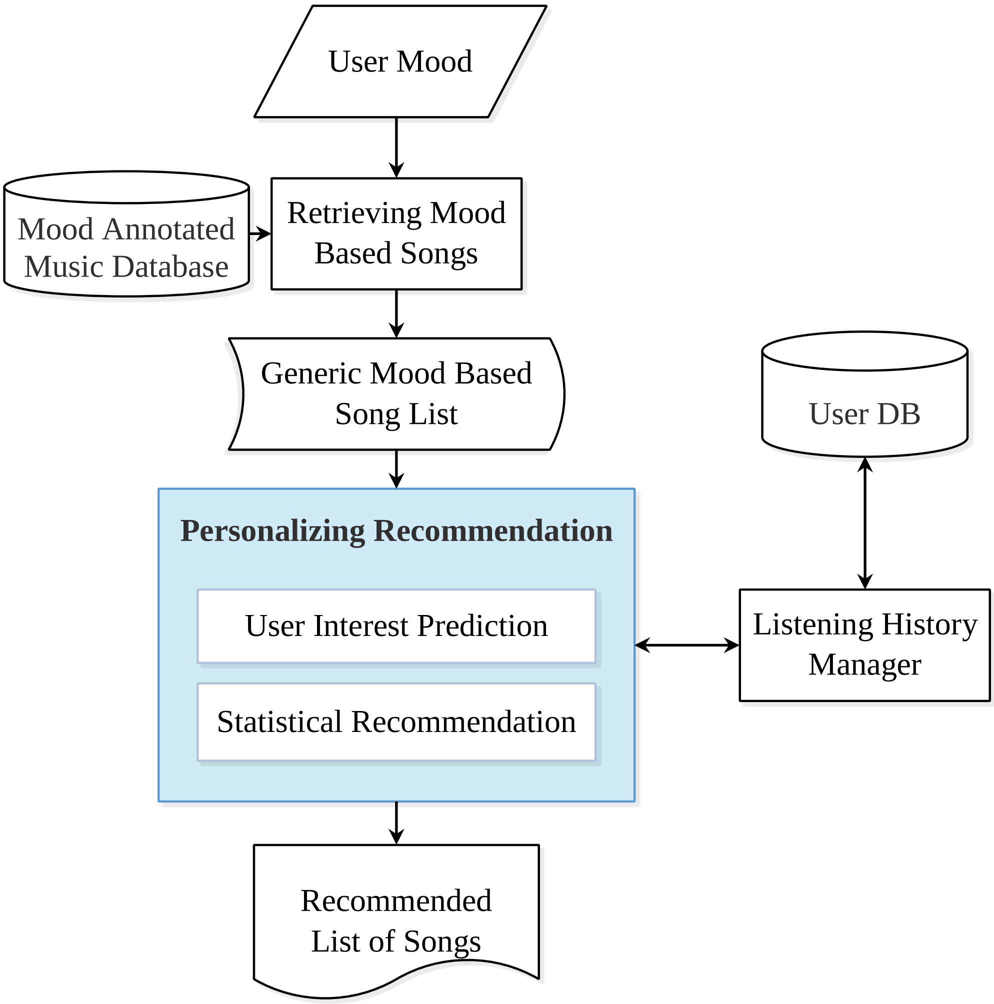


Figure 21*: The recommender component*

Then generic mood based list of songs that matches with the user mood are retrieved from a music database (Algorithm 6 line 1). These list of songs are first-round candidates that are not last recommendation, which need to be refined. They are selected only by associating the user mood with the audio song features. That need to be personalized/filtered according to the user history or personal preference.

4.6.2 Listening History Manager

The listening history manager component track user's listening experience and manages the user listening history. It frequently updates user database to make the recommendations more personalized. This component collects recommended song's information from the recommender. Post recommendation information also collected from the user smartphone. These include: 1) Song information that is frequently ignored or listened by the user, which are assigned a value 1 for listened and 0 for ignored songs; 2) how the user reacted for the songs in tags or comments. These tags or comments are analyzed to label the song with respect to the user personal perception.

4.6.3 Personalizing Recommendation

This module accepts first-round candidate songs and personalizes it based on the listening experience of the user. The recommendation list should be small and personalized so that the generic list is reduced by making it more personalized associating it with the user listening history. To do this, two conditions are checked from the listening history of the user that are accessed via listening history manager component. These conditions are stated as follows:

- The first condition is if the user listened to the recommend songs or not which is represented by 1 and 0. If the user frequently ignored a song, the song will be registered as not listened and the listened variable in the user profile/history is assigned 0 value else it will be assigned 1 to mean the user has listened to the recommended song.
- The other condition is if the recommended songs match with the user personal perception or not. These are known if the user tagged the recommended song and this tag match or not with the intended music mood class by the system.

Considering these two conditions the songs that are not listened and those that do not match with the individual perception are removed.

User Interest Prediction: most frequently listened and/or songs that are tagged by adjective that indicate user interest are suggested as user's interest. Then similar songs from the first round candidates that are thought to be liked by the user are nominated for the recommendation. Due to the fact, that list of filtered songs in the first round candidate are small in number as well as the feature parameter values are numeric data, we proposed the Euclidean distance measure (Equation 20) to filter those similar songs (Algorithm 6 line 15-19).

$$dist = \sqrt{\sum_{k=1}^{n}(p_k - q_k)^2} \qquad \textbf{\textit{(20)}}$$

Where:

- n is the number of music features
- p_k and q_k are the k^{th} features of songs p and q respectively

The distance obtained between previously liked songs and new songs determines how to rank recommended list of songs. The lower the distance indicates the higher the similarity so that those songs which have lower distance are likely to be recommended.

Statistical Recommendation: in order to prevent too much similarity and repetition of recommended songs. We combined songs that are not previously listened/less listened nor rejected by the user. This is performed using statistical information from the user listening history (Algorithm 6 line 21-28).

Finally, recommendations from the user interest prediction and statistical recommendation creates filtered list of recommended songs that contains previously listened and new songs. Based on this data, the recommender generates final recommendation and updates (Algorithm 6 line 29-31).

***Algorithm** 6: Recommender*

```
Input: user_mood,
    listening_history[]
Variable: i = 0, j = 0
Output: recommended_song[],
    updates[]
Begin
 1. songs4currentMood[] = MusicDB.getSongs (user_mood)
 2. prev_recom[]= History_Manager.getRecomList(user_mood)
 3. do{
      If(prev_recom[i].listened = 0 & prev_recom[i].tags
    ≠ music_mood){
 4.             rejected[j] = prev_recom[i].song_id
 5.             j = j+1
 6.        }
 7.      i = i +1
 8. }while(count(prev_recom[]) < i)
 9. foreach(songs4currentMood[]){
10.        if(songs4currentMood[] ≠ rejected[]){
11.    temp[] = songs4currentMood[]
12.   }
13. }
```

```
candidate_songs[] = temp[]
interestBasedPred(candidate_songs[], prev_recom[]){
  liked_songs[]=intersect(candidate_songs[], prev_recom[])
  similar_songs[]=euclideanDistanc(candidate_songs[].features, liked_songs[].features)
  interest_based_songs[]=orderBy(similar_songs[].distance, ascending)
  return interest_based_songs[]
}
statisticalRecom(candidate_songs[], prev_recom[]){
  songs_listened[]=intersect(candidate_songs[], prev_recom[])
  now_recom[] = interestBasedPred()
  while(songs_listened[] ≠ candidate_songs[] & now_recom[] ≠ candidate_songs[]){
    new_songs[] = candidate_songs[]
  }
  return new_songs[]
}
recommended_song[] = interest_based_songs[]
recommended_song[] = append.new_songs[]
updates[] = insert (recommended_song[], user_mood)
End
```

Chapter 5: Experimentation and Evaluation

This chapter covers the experiments and the evaluation methods, including the results found, which is detailed in four main parts. These are 1) Experimental setup, discussing acquisition of data and tools used in Section 5.1; 2) Proposed System Implementation as detailed in Section 5.2; 3) Results found as discussed in Section 5.3; and 4) Overall discussion of the proposed system in Section 5.4.

4.7 Experimental Setup

5.1.1 Acquisition of Data

Music Data: in the study of Selam Taddesse [1] and in the preliminary study we conducted on music mood or related fields of music psychology, there is no music corpus or any collection of musical data that illustrate or demonstrates Ethiopic music with respect to mood. Therefore, it was necessary to collect and manually annotate songs to prepare music corpus that contain defined set of mood required to train the MMR model. We conducted offline as well as an online survey to get annotated set of music data. The survey has 600 songs and annotated by 54 randomly selected participants. The songs are collected by three people from different sources to make the collection more balanced. The main sources of songs include: music streaming websites such as Zemabet[15] and AddisZefen, social media YouTube[16] and Facebook[17]. Others sources (e.g. radio, lyrics websites) are also used as a direct source or as a reference to find diversified set of music collection. The corpus preparation task includes audio song segmentation and manual annotation of a set of music collection.

Segmentation: the songs collected for this study are from different sources that are full length and some of them have noise (e.g. advertisements). Each songs are segmented into 30 seconds excerpts which are prepared for the manual annotation and audio feature extraction. In addition to this, the noise contained in some of the songs are trimmed out and those songs, whose audio format is not found are converted into an audio format from their video clip.

Annotation: in order to get mood based categorized music data, doing manual annotation is comparatively the easiest way in our case. Therefore, we prepared a survey to annotate the songs based on the Tellegen, Watson, and Clark customized categorical music mood model (Section 4.2.1.).

[15] https://zemabet.com/
[16] https://youtube.com/
[17] https://facebook.com/

The survey distributed to the participants both offline and online. For the purpose of offline survey a Microsoft Excel is used, in which the audio songs linked with their corresponding information and list of tags. Sample of the offline survey questionnaire is presented in Annex A. Online survey is conducted using Google form. At the time of this survey Google form doesn't support uploading audio file. Therefore we converted 600 audio excerpts into video. Then these videos are uploaded on YouTube and embedded in the Google form. Sample of the online survey questionnaire is presented in Annex B.

Both the offline and online survey has 30 different forms (questionnaires) each containing 20 distinct song excerpts. Among the 54 participants of the survey, 15 are female and 39 are male with age between 19 and 45. The subjects are instructed to listen to the songs at least for 30 seconds (full of the excerpt at least once). After listening to the song, the person tags it with the mood she/he perceived using the adjectives provided in the music mood model.

Table 5*: Summary of annotated music mood*

Mood Class	**Tags**	**Songs**	**Songs (%)**
HPA	active, elated, enthusiastic, excited, peppy, strong	136	22
LNA	at rest, calm, placid, relaxed	112	19
P	content, happy, kindly, pleased, satisfied, warmhearted	78	13
SE	aroused, astonished, surprised	72	12
UnP	blue, grouchy, lonely, sad, sorry, unhappy	121	20
LPA*	drowsy, dull, sleepy, sluggish	10	2
HNA*	distressed, fearful, hostile, jittery, nervous, scornful	17	3
DisE*	quiescent, quiet, still	5	1
Multiple		49	8

A * in Table 5 indicates set of mood categories rejected in this study

Generally, the annotated set of songs are grouped into the eight primary mood types according to the vote of majority. Some of the songs that are tagged into multiple mood classes (primary moods) are rejected. Three categories of mood (LPA, HNA, and DisE) also has got 17 and less than 17 songs. So, these three classes are not good enough to represent the mood of the subjects in the case of our study, so that they are rejected including the songs categorized into these classes. Table 5 summarizes the result of the annotation.

Heart-Rate and Text Data: in order to construct the user mood detection component, we need heart-rate and mood sentence training data. Heart-rate data is collected from expert's experimental results as indicated in Section 4.2.3.1.

A set of annotated mood sentences are collected from different data repositories. These include, Data World[18], Parallel Dots[19], WWBP[20], Roman Klinger[21] and Github[22]. The combined set of annotated sentences contains a total of 25,800 sentences. Each type of mood contains 5,160 sentences.

5.1.2 Tools

In order to justify the proposed recommender, a prototype is built using tools and programming languages listed in Table 6. Here we mentioned the most important technologies which include hardware, software, operating system and programming languages.

Table 6*: Tools and Programming languages used for experimentation*

Tools and Programming languages		**Specification**
Hardware	RAM	4 GB
	CPU	Intel Pentium Cori3
	Hard disk	500 GB
Application Software	Anaconda	Version 3-4.3.1
	MATLAB	Version R2017a
	MIR Toolbox	Version 1.7
	Audacity	Version 2.1.0
	NLTK's (Tokenizer, Stop Word removal)	
	Gensim	
Programming Language	Matlab	R2017a
	Python	Python 3
Operating System	Windows	Win 7 Ultimate, 64bit

[18] https://data.world/crowdflower/
[19] https://blog.paralleldots.com/product/emotion-detection-using-machine-learning/
[20] http://wwbp.org/downloads/public_data/dataset-fb-valence-arousal-anon.csv
[21] http://www.romanklinger.de/ssec/
[22] https://github.com/huseinzol05/NLP-Dataset

Audacity: is a free open source digital audio processing tool. It is used to preprocess audio songs like trimming and noise removing before feature extraction stage.

MIRtoolbox: is a MATLAB toolbox for musical feature extraction from the audio song. It is a project of "Tuning the Brain for Music" under the European context. The software is published by the Free Software Foundation under the terms dedicated to the study of music and mood. We have used MIRtoolbox 1.7 (03, June 2017 release) for audio music feature extraction and analysis.

MATLAB: is a proprietary visualization, programming and numerical computing environment developed by MathWorks. We used it to implement music mood classification algorithms and plotting of functions and data. MATLAB also allowed us to install the MIRtoolbox as a module in its environment.

In addition to MATLAB programming language, parts of the system are implemented using Python programming language. These parts of the system include the context manager and recommender components.

Different NLP tools are also used for the pre-processing and training phases of the context manager component of the proposed recommender. The main tools include NLTK's module (Tokenizer and Stopwords Removal) and Gensim. NLTK's Tokenizer is used to split sentences into words to create list of tokens. This is latter used to build a dictionary from the sentences that gives unique id for all words. The NLTK's Stopwords module used to filter a set of stop words from the sentences to be processed, which contains 153 English stop words. Gensim module is used to build a model that computes weighted score of sentences using TF-IDF space.

4.8 Implementation

MMR Model: in order to construct best performing model we conducted an experiment using four algorithms such as Subspace Discriminant, Linear Discriminant, Cosine KNN, and Linear SVM. In the Experiment, the Linear SVM algorithm showed best performance compared to the other tested algorithms. Therefore, the music mood recognition model is constructed using linear SVM. The interface of the music classifying model in Figure 22 shows an instance of mood recognition of tizita.waw, which is annotated as Unpleasantness with mood tags including lonely, sad and sorry.

Context Manager: in order to predict current user's music taste, user information is collected in different ways. As it's depicted in Figure 23, mood based query sentences and heart-rate data can be provided to draw list of recommended songs. In addition to user

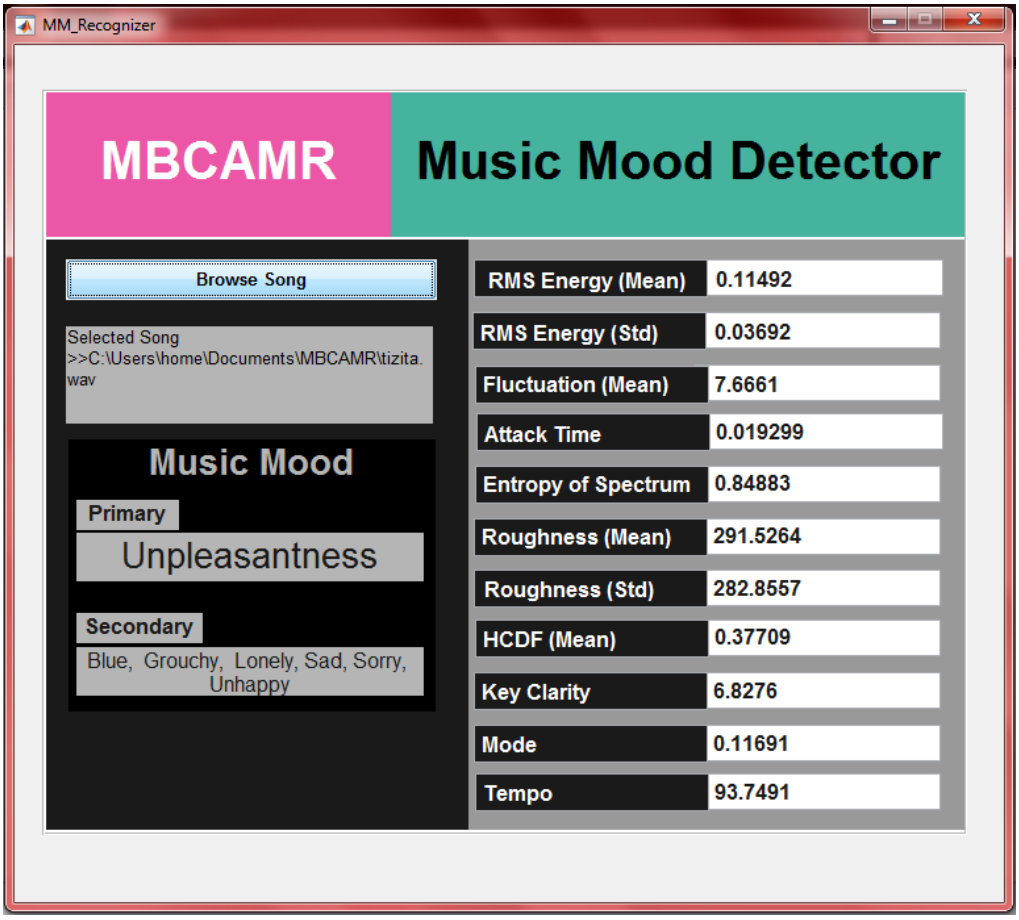


***Figure 22**: Music Mood Detection interface*

query, heart-rate sensor and user listening experience are used to make suggestions automatically.

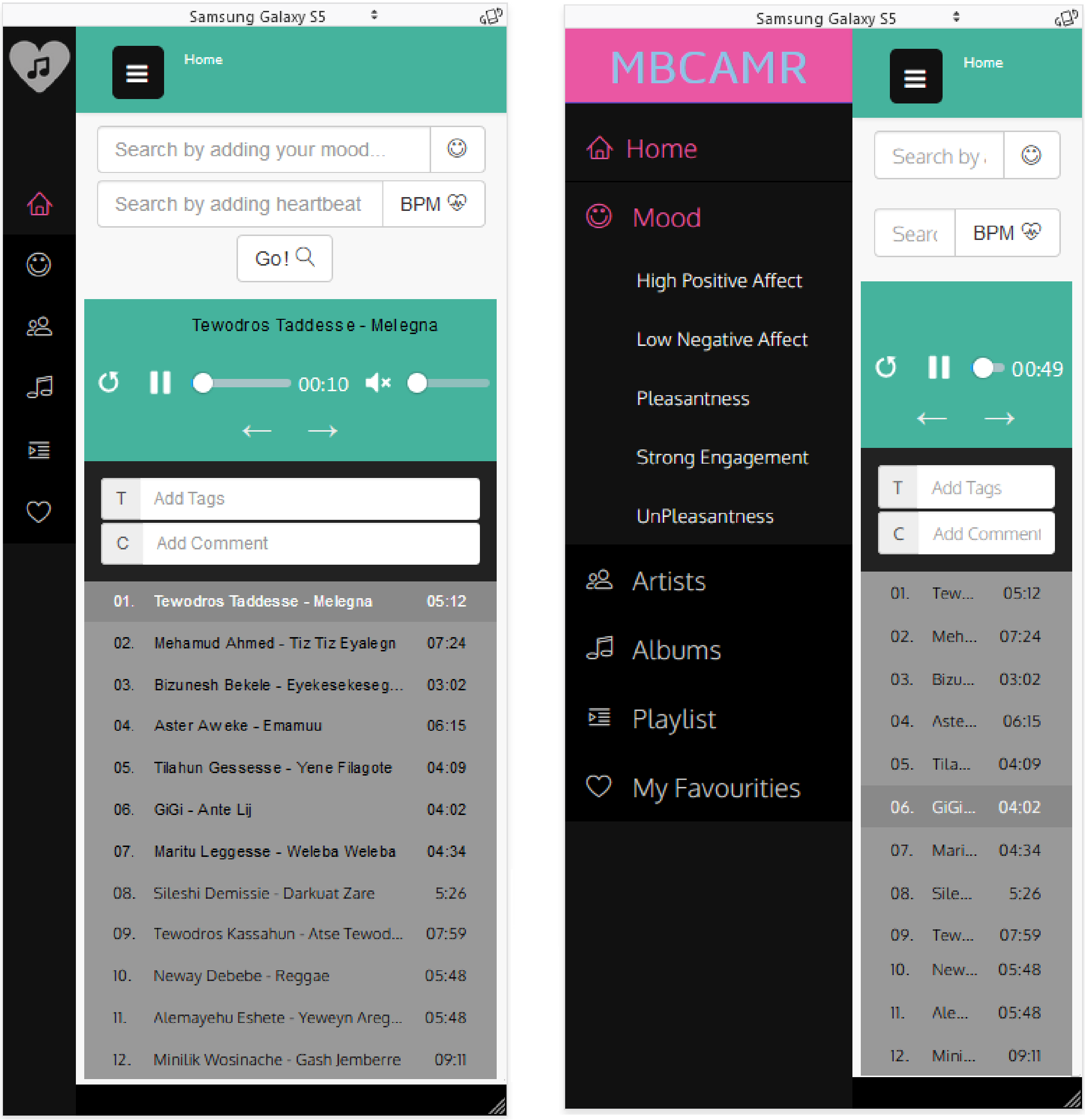


***Figure 23**: Mood based music exploring options*

Recommender: the recommender generates randomly selected generic mood based list of songs for first time users. Then personalized mood based recommendation is made while the users are using the system. Figure 24 and 25 shows screenshots of an instance of the recommendations generated respectively. Figure 24 depicts an instance of randomly generated mood based songs for first time user with a state of mood such as active, elated, enthusiastic, excited, peppy, and strong. Figure 25 depicts an instance of mood based personalized music recommendation containing: 1) Most frequently listened songs in a situations currently the user is found; 2) New or rarely listened songs recommended based on user listening history /statistical data; 3) Songs that are not listened previously but similar to most frequently played songs. These doesn’t included those songs that are frequently recommended but ignored.

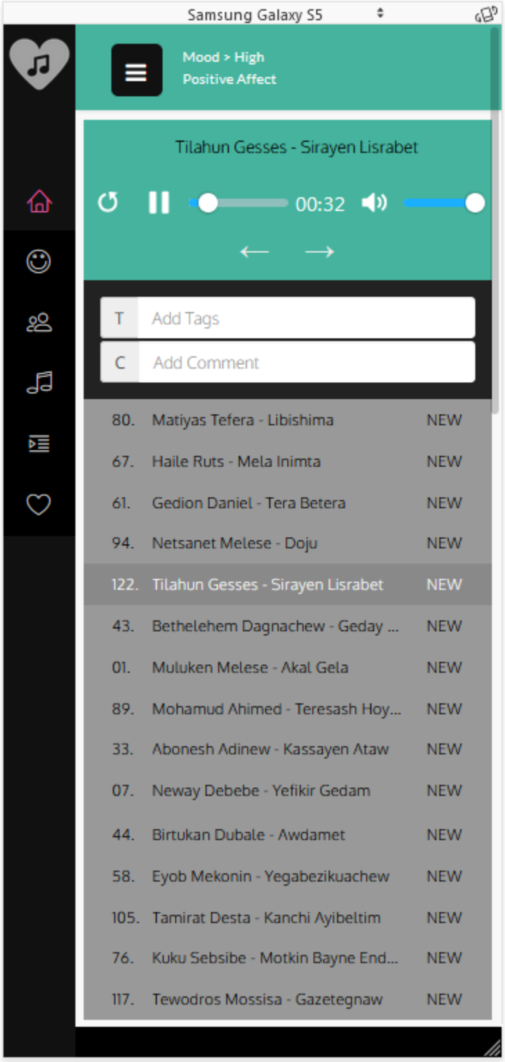


***Figure 24**: Generic mood based recommendation for first time users*

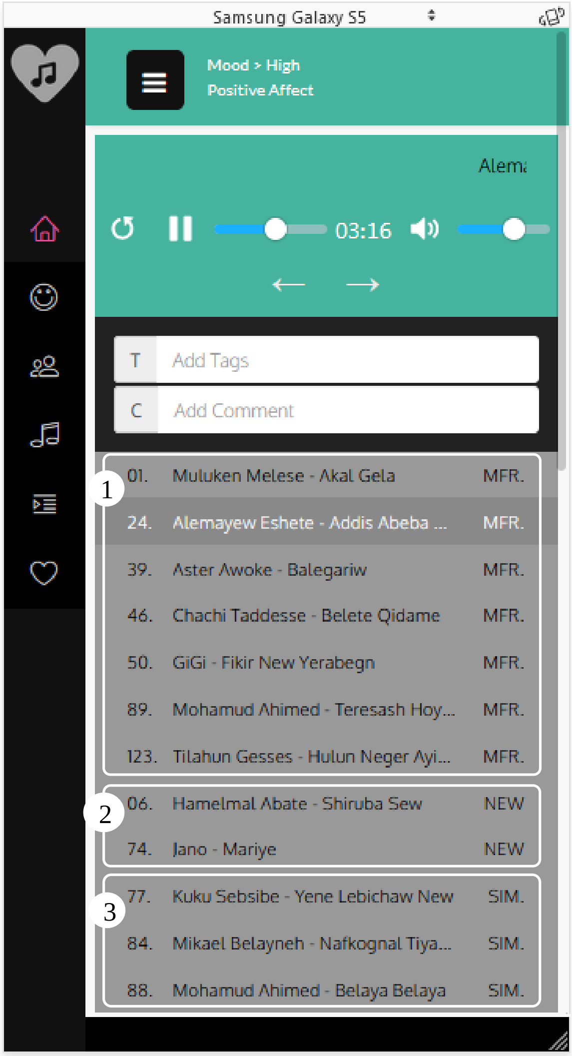


***Figure 25**: Personalized mood based recommendation*

4.9 Results

MMR Model: music mood recognition model is constructed using Linear SVM after comparing it with other three algorithms, such as Subspace Discriminant, Linear Discriminant, and Cosine KNN. All of the algorithms are evaluated based on cross validation in which k = 10. The validation is repeated at least five times to test the algorithms using a different set of test data. Table 7 presents summary of the performance of the four algorithms on each type of music mood.

***Table 7**: True positive rate of tested algorithms*

	Linear SVM (%)	**Cosine KNN (%)**	**Linear Discriminant (%)**	**Subspace Discriminant (%)**
HPA	59	41	67	67
LNA	37	45	9	9
P	50	6	50	19
SE	60	57	57	43
UnP	51	71	58	63
Overall Accuracy	51.5	45.6	48.5	42.7

Using only appropriate sets of features, as presented in Table 5, in each pair of classes and standardizing the feature values can improved accuracy of the music mood classification model. Therefore, by applying this modification, the Linear SVM gives over all accuracy of 65%. The confusion matrix (Figure 26) illustrates the true positive and false negative rate of the Linear SVM classifier for each type of music mood.

True Class \ Predicted Class	HPA	LNA	P	SE	UnP	True Positive Rate	False Negative Rate
HPA	84%	6%	3%	2%	5%	84%	16%
LNA	11.5%	71%	5%	3%	9.5%	71%	29%
P	17.5%	12.5%	43%	16%	11%	43%	57%
SE	13.5%	7%	6.5%	55%	18%	55%	45%
UnP	10%	11.5%	3.5%	3%	72%	72%	28%

***Figure 26**: MMR model confusion matrix*

Context Manager: user mood detection as part of user's music test prediction performed using heart-rate and text. The histogram (Figure 27) illustrates result of an experiment that compares sentence based user mood detection and the DS theory based user mood detection approaches. As stated in Section 4.2.4 the DS theory approach incorporates sentence and heart-rate data to detect user mood.

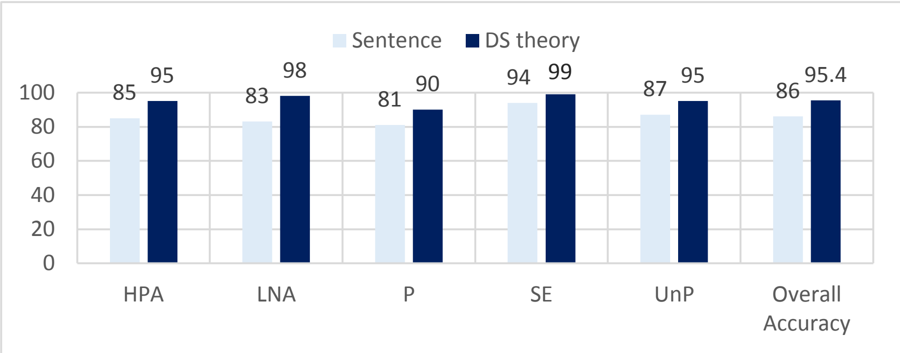


***Figure 27**: Comparison of user mood detection using sentence and DS theory*

The confusion matrix (Figure 28) also depicts accuracy of the proposed system, showing the true positive and false negative rate of the user mood detection module.

True Class	Predicted Class: HPA	LNA	P	SE	UnP	True Positive Rate	False Negative Rate
HPA	95%	10%				90%	10%
LNA	1%	98%		1%		98%	2%
P	3%	6%	90%	1%		90%	10%
SE			0.6%	99%	0.3%	99%	1%
UnP		4%		1%	95%	95%	1%

***Figure 28**: User Mood Detection evaluation confusion matrix*

4.10 Discussion

This study has shown one way of Ethiopic music recommendation. As it's the first mood based recommender focusing on Ethiopic song, the results obtained can be considered as good, but it has also a lot of defects which require further study. In order to evaluate and test the proposed system, we have used different evaluation methods for individual components and overall system.

MMR Model: there is no previous work related to music mood classification for Ethiopic music. Other music mood dataset with different culture is also not found to test and compare the proposed algorithm. Therefore, the performance of the proposed MMR model is evaluated by comparing the performance of other commonly known best performing algorithms. These includes Subspace Discriminant, Linear Discriminant, Cosine KNN and Linear SVM whose result is shown in Table 7.

Subspace Discriminant: The MMR model based on Subspace Discriminant algorithm has overall accuracy of 42.7%. That classifies nearly 43% of the songs correctly and 57% of the songs are assigned to wrong music mood class. Accuracy of the Subspace Discriminant algorithm for each type of music mood is presented in Table 7. Among the five types of music mood, the HPA and UnP songs have the highest classification accuracy of 67% and 63% respectively. However, LNA and P category of songs have the lowest accuracy of 9% and 19% respectively. Accuracy of Subspace Discriminant can be considered good for HPA and UnP, it is hardly so for SE (43%). But this algorithm shows poor performance for LNA and P type of songs.

Linear Discriminant: the Linear Discriminant algorithm comparatively performs well with respect to the Cosine KNN and Subspace Discriminant algorithms. It has overall accuracy of 48.5%. As presented in Table 7, the Linear Discriminant performs well for HPA (67%), P (50%), SE (57%), and UnP (58%). However, similar to the Subspace Discriminant algorithm, the Linear Discriminant algorithm also shows very poor accuracy for LNA (9%).

Cosine KNN: experimenting using the Cosine KNN algorithm gives very good accuracy for UnP (71%) songs with k value of 10. However, it has overall accuracy of 45.6%, which is poor compared to accuracy of the Linear SVM and Linear Discriminant algorithms. Cosine KNN algorithm performance is very poor in P type of songs, it classifies 94% percent of the songs into a wrong type of music mood. This can be considered that, the Cosine KNN algorithm is not applicable to classify songs of Ethiopia that induces pleasantness, with the features extracted for this experiment.

Linear SVM: in the experiments we conducted, the Linear SVM classifier gives more robust and fast results for discriminating among the music mood classes. With respect to the other algorithms, the Linear SVM algorithm performs best, with overall accuracy of 51.5%. Then making some modification as stated in Section 5.3, the accuracy improved to 65%. This is the best result in music mood classification using only audio features.

The confusion matrix (Figure 23) illustrates the true positive and false negative rate of the Linear SVM classifier for each type of music mood. From the results found, the proposed algorithm (Linear SVM) outperforms the other three algorithms (Cosine KNN, Subspace Discriminant and Linear Discriminant). As it's illustrated in the confusion matrix, the Linear SVM algorithm shows good performance on four of the music types, and hardly so for P type of songs. Therefore, the music mood recognition model is constructed using linear SVM.

Context Manager: one of an interesting research question in this study was whether incorporating different sources of user context information increases confidence in the system during user mood detection. From the result found, incorporating different contextual data for user modeling of recommendation system has a significant improvement. As illustrated in the histogram (Figure 27), the proposed user mood detection module of the user modeling component has a significant improvement of 9.4% over a sentence only context based user mood detection. Generally the proposed DS theory showed good performance for all type of moods.

The recommender: in addition to the evaluation of individual components of MBHEMR, the recommender model is generally tested by 10 volunteer participants who are instructed to evaluate the proposed model. Based on the feedback from the participants the proposed recommender generates more convincing recommendations.

The evaluation was made by comparing user expectation and the generated results. The users are asked to add context information such as heart-rate and textual description of their current mood (as depicted in Figure 23 that shows screenshot of front page of the recommender). Subjects listen to songs recommended for current mood detected by the system. Then they give feedback for the generated recommendation whether they agree or not with the established match between the recommended songs and their mood. From the response given by the participants, the proposed recommendation model gives a promising result.

Chapter 6: Conclusion and Future Work

6.1 Conclusion

Music plays an important role in people's daily life. It affects heart-rate and has benefit for mental health. Music captivates large amount of Internet users and an enormous amount of music is being produced. Among these, some music succeeded in attracting attention of millions of users, while some others critical for current user mood remain obscured.

Recommendation systems can alleviate these and other related problems by associating music and personal data. Music mood perception is affected by cultural and listening background of humans. However, there is no mood based music recommendation system for Ethiopian culture and listening habit. Although there are already different ways to draw up recommendations, users are still not satisfied. There are still demands for services that support music navigation, discovery, and sharing.

The characteristics of music content as well as variety in perception of music mood in humans makes the process of building mood based music recommendation system complex and challenging. Generally music recommender system has three main components, such as music item modeling, user modeling and music-user matching components. These components can be combined using different approaches including metadata, collaborative, content-based, context-aware, and hybrid filtering, each of them with their own pros and cons.

The music item modeling component involves music mood detection process that requires choosing appropriate music mood model. Categorical and dimensional models are the two main music mood modeling concepts to define the possible set of moods conveyed by a song. User modeling also requires combination of different contextual information in order to match the music item with users' preference.

In the previous studies researchers proposed recommendation systems that are ought to minimize user effort and narrow choices to satisfy users need. However, current systems still require a lot of user's effort. Some of them determine user preference based on others preference rather than their own perception. Music content or features that can influence the human state of mood need to be studied with respect to specific culture and listening experience; User mood as context parameter need further study.

Among the previous studies, recommender systems that are based on the CF approach require more time and data to draw appropriate recommendations. CF approach based works also suffer from popularity bias. On the other hand, works that are based on the CBF approach usually give obvious recommendations due to their content similarity criteria. To fill gaps in the individual approaches, a lot of researcher communities in this domain suggests to hybridize these approaches to get the best out of them.

In this research, we designed and implemented mood based context aware music recommendation system for smartphones. To realize this different methodologies are applied including relevant literature review, collecting music and user related data, using helpful tools for experimenting with the proposed solutions and evaluation.

The proposed MBHEMR in this research has three main tasks. The first task is classifying Ethiopic music into different mood types. For this, we need a music corpus but there is no previously available mood based music data. Therefore, we have prepared a music mood corpus that contains 11 selected best discriminating audio song features extracted from 522 songs which is manually annotated by 54 survey participants. The corpus is then used to train MMR model using linear SVM.

The other main task of this research was user modeling that involves user mood detection from contextual data. In order to detect the user mood, we combined biometrics (heart-rate) and text modalities of mood expression. As the context data are from separate independent sources, we developed a new method to combine them using the DS theory that enables as handling uncertainties in user mood prediction.

Once the music item modeling and user modeling is completed, we have created a recommender component (an algorithm that create association between the user interest and songs, to draw list of recommended songs). The recommender component draws personalized recommendations based on user interest and statistical data. User interest is computed from their listening experience through tags, comments and frequency of a song listened. In order to prevent too much similarity in the recommendation, statistical recommendation is performed using statistical information from the user listening history. This combines new songs that are not previously recommended to the user with previously liked songs.

The proposed MBHEMR is justified using a prototype. The music mood classification component is built on Linear SVM, after experimenting and comparing it with Subspace

Discriminant, Linear Discriminant, and Cosine KNN based algorithms. These algorithms are compared by their accuracy of classification, in which Subspace Discriminant with accuracy of 42.7%, Linear Discriminant with accuracy of 48.5%, Cosine KNN with accuracy of 45.6% and Linear SVM 65% of accuracy. From the results found, the proposed algorithm (Linear SVM) shows a better performance.

The user mood detection module of the user modeling component is constructed by incorporating heart-rate data collected from expert's experimental results and 25,800 sentences from different sources that are correspondent with the music mood model. The result shows that incorporating different contextual data for user modeling of recommendation system has a significant improvement of 9.4% over a sentence only context based user mood detection. Generally the proposed DS theory method has overall accuracy of 95.4%, while the user mood detection module using only text (sentences) got 86% of accuracy. The MBHEMR is generally tested by volunteers to evaluate the proposed model. Form the feedbacks found the proposed system generates suitable recommendations.

Contribution

Previously music recommenders are proposed to create better listening experience. The system we proposed makes mood based music recommendation, which gets a good feedback from users. Generally, the contribution of this thesis are presented as follows:

- The study revealed a method to classify Ethiopic songs
- A new way of user modeling that track and detect user mood using different independent sources of information, which can be easily extended with additional source of information.
- Algorithm for automatic music mood annotation and fusing different sources of data.
- A prototype of context aware mood based music recommender.
- A new music mood dataset containing appropriate features, which can be used in a related studies.

6.2 Future Work

All in all the proposed work contributed a lot for the research area. The results obtained can be considered as a very promising result. However, a lot of works can be done to improve performance the proposed system. These include:

- Improving the accuracy of the music mood detection component by adding more ground truth data that participates additional user community.
- Adding more music features in addition to audio feature parameters can also make the recommendation more semantic. These might include lyrics, genre, artist, track name, social tags/comments etc.
- Applying the DS theory in the other components like the music mood detection component to improve user trust and satisfaction on the recommender system.
- Exploiting already available additional sensor data also can be used to reduce efforts of the user (e.g. accelerometer sensor data).
- Considering Amharic query sentences in the context manager to improve user's music taste prediction.

References

[1] S. Taddesse, "Knowledge and Attitude of Profesionals Towads Music Therapy in Addis Ababa: The Case of Health Professionals Working at St. Yared General hospital," Addis Ababa, june 2011.

[2] L. Eyre, "Music Therapy in Mental Health: Practice, Theory, Research, and Professional Perspectives," The American Music Therapy Association, Advanced access publication, 2015.

[3] MS Cepeda; DB Carr; J. Lau; and H. Alvarez, "Music for Pain Relief," Cochrane Database of Systematic Review 2, 2006.

[4] M. Clark; G. Isaacks-Downton; N. Wells; S. Redlin- Frazier; C. Eck; JT. Hepworth; and B. Chakravarthy, "Use of Preferred Music to Reduce Emotional Distress and Symptom Activity During Radiation Therapy," *Journal of Music Therapy,* p. 43, 2006.

[5] "Music and Mood," 21 November 2015. [Online]. Available: https://www.healthychilderen.org/english/healthy-leaving/pages/music_and_mood.asp. [Accessed 6 Feb 2017].

[6] http://www.livestrong.com/article/75323-relationship-between-music-heart-rate/, "Relationship Between Music and Heart Rate," LIVESTRONG, 19 Aug 2013. [Online]. [Accessed 9 Feb 2017].

[7] Ding Yiwen; and Liu Chang, "Exploring Drawbacks in Music Recommender Systems – The Spotify Case," University of Boras, 2015.

[8] Kosta Katerina; Song Yading; Fazekas Gyorgy; and Sandler B. Mark, "A Study of Cultural Dependence of Perceived Mood in Music," in *International Society for Music Information Retrieval*, Centre for Digital Music, Queen Mary University of London, 2013.

[9] "Music of Ethiopia," 11 Jan 2017. [Online]. Available: https://en.wikipedia.org/wiki/Music_of_Ethiopia. [Accessed 9 Feb 2017].

[10] E. Abate, "Ethiopian Kiñit (scales) Analysis of the Formation and Structure of the Ethiopian Scale System," in *16th International Conference of Ethiopian Studies, ed. by Svein Ege, Harald Aspen, Birhanu Teferra and Shiferaw Bekele, Trondheim*, 2009.

[11] M. Kaminskas and F. Ricci, "Contextual music information retrieval and recommendation: State of the art and challeges," Elsevier Inc., Bolzano, Italy, 2012.

[12] Jae Sik Lee; and Jin Chun Lee, "Music for My Mood: A Music Recommendation System Based on Context Reasoning," Springer-Verlag , Berlin Heidelberg, 2006.

[13] U. Shardanand and M. P, "Social information filtering: algorithms for automating “word of mouth”," in *SIGCHI Conference on Human Factors in Computing Systems, ACM Press/Addison-Wesley Publishing*, New York, NY, USA, 1995.

[14] S. Reddy; and J. Mascia, "Lifetrak: Music in Tune With Your Life," in *the 1st ACM International Workshop on Human-Centered Multimedia*, New York, NY, USA, 2006.

[15] Martin Pichl; Eva Zangerle; and Gunther Specht, "Towards a Context-Aware Music Recommendation Approach: What is Hidden in the Play list Name?," in *IEEE 15th International Conference on Data Mining Workshops*, 2015.

[16] Han-Saem Park; Ji-Oh Yoo; and Sung-Bae Cho, "A Context Aware Music recommendation System Using Fuzzy Bayesian Networks With Utility Theory," Springer-Verlag, Berlin Heidelberg, 2006.

[17] Chengkun Jiang; and Yuan He, "Smart-DJ: Context-Aware Personalization for Music Recommendation on Smartphones," in *IEEE 22nd International Conference on Parallel and Distributed Systems*, China, 2016.

[18] "Music Emotion Recognition by Multi-Label Multi-Layer Multi-Instance Multiview," in *ACM International Conference on Multimedia*, 2014.

[19] F. FIGUEIREDO, J. M. ALMEIDA, M. A. G. ALVES and a. F. BENEVENUTO, "On the Dynamics of Social Media Popularity: A YouTube Case Study," ACM Transactions on Internet Technology, Universidade Federal de Minas Gerais, Brazil, 2014.

[20] Yading Song; Simon Dixon; and Marcus Pearce, "A Survey of Music Recommendation Systems and Future Perspectives," in *9th International Symposium on Computer Music Modelling and Retrieval (CMMR 2012)*, Queen Mary University of London, 19-22 June 2012.

[21] Peter Brusilovsky; Alfred Kobsa; and Wolfgang Nejdl, "Methods and Strategies of Web Personalization," Springer-Verlag , Berlin Heidelberg , 2007.

[22] Prem Melville; and Vikas Sindhwani, "Recommender Systems," IBM T. J. Watson Research Center, Yorktown Heights, NY, USA, 2017.

[23] Gediminas Adomavicius; Bamshad Mobasher; Francesco Ricci; and Alex Tuzhilin, "Context-Aware Recommender Systems," Association for the Advancement of Artificial Intelligence, 2011.

[24] R. J. Mooney, "Content Based Book Recommending Using Learning for Text Categorization," in *Appears in Proceedings of the SIGIR-99 Workshop on Recommender Systems: Algorithms and Evaluation*, Berkeley, CA, August 1999.

[25] F. Pachet, "Knowledge Management and Musical Metadata," in *in Encyclopedia of Knowledge Management, Schwartz, D. Ed. Idea Group* , 6 rue Amyot, 75005 Paris , 2005.

[26] J. S. Downie, "Music Information Retrieval," in *Annual Review of Information Science and Technology*, 2003.

[27] John S. Breese; David Heckerman; and Carl Kadie, "Empirical Analysis of Predictive Algorithms for Collaborative Filtering," Microsoft Research , Redmond, WA 98052-6399 , 1998.

[28] J. .. Rocchio, "Relevance Feedback in Information Retrieval," 1965.

[29] A. K. Dey, "Providing Architectural Support for Building Context-Aware Applications," Georgia Institute of Technology , 2000.

[30] Linas Baltrunas; Marius Kaminskas; Bernd Ludwig1; Omar Moling1; Francesco Ricci; Aykan Aydin; Karl-Heinz Luke; and Roland Schwaiger, "InCarMusic: Context-Aware Music Recommendations in a Car," Springer-Verlag, Berlin Heidelberg, 2011.

[31] SUSAN HALLAM; IAN CROSS; and MICHAEL THAUT, The Oxford Handbook of Music Psychology, Great Clarendon Street, Oxford, OX2 6DP, United Kingdom: Oxford University Press, 2016.

[32] V. J. Konecni, The Psychology of Music - Social Interaction and Musical Preferece, Acadamic Press INC (Harcour Brace Jovanovich, Publishers), 1982.

[33] Adrian C. North; and David J. Hargreaves, "Situational Influences on Reported Musical Preference," Psychomusicology, University of Leicester, 1996.

[34] R. Burke, "Hybrid Web Recommender Systems," Springer, Verlag Berlin Heidelberg , 2007 .

[35] Youngmoo E. Kim; Erik M. Schmidt; Raymond Migneco; Brandon G. Morton; Patrick Richardson; Jeffrey Scott; Jacquelin A. Speck; and Douglas Turnbully, "Music Emotion Recognition: A State of The Art Review," in *11th International Society for Music Information Retrieval Conference* , ISMIR, 2010.

[36] William Forde Thompson; and Lena Quinto, "Music and Emotion: Psychological Considerations," in *The Aesthetic Mind: Philosophy and Psychology*, Oxford, Oxford University Press, 2011, pp. 357-375.

[37] R. Cowie; E. Douglas-Cowie; K. Karpouzis; G. Caridakis;M. Wallace; and S. Kollias;, "Recognition of Emotional States in Natural Human-Computer Interaction," IEEE, 2009.

[38] Peter Dunker; Stefanie Nowak; André Begau; and Cornelia Lanz, "Content-based Mood Classification for Photos and Music: A Generic Multi-modal Classification Framework and Evaluation Approach," ACM, Vancouver, British Columbia, Canada, 2008.

[39] Yi-hsuan Yang; and Homer H. Chen, "Machine Recognition of Music Emotion: A Review," ACM, National Taiwan University, 2012.

[40] P. R. Farnsworth, "A Study of the Hevner Adjective List," *The Journal of Aesthetics and Art Criticism,* Vols. Vol. 13, No. 1, pp. 97-103, 1954.

[41] David Watson; David Wiese; Jatin Vaidya; and Auke Tellegen, "The Two General Activation Systems of Affect: Structural Findings, Evolutionary Considerations, and Psychobiological Evidence," *Journal ol' Personality and Social Psychology,* Vols. Vol. 76. No. 5, 820-K38, pp. 820-838, 1999.

[42] R. W. Picard, "Affective Computing," MIT Media Laboratory; Perceptual Computing; 20 Ames St., Cambridge, MA 02139, 2000.

[43] J. A. Russell, "A Cirmuplex Model of Affect," *Journal of Personality and Social Psychology,* vol. 39 No. 6, pp. 1168-1171, 1980.

[44] J. Grekow, "Representations of Emotions," in *From Content-Based Music Emotion Recoginition to Emotion Maps Musical Pieces*, Springer International Publishing, 2018, p. Chapter 2.

[45] M.V. Mathews; I. Bengtsson; A. Gabrielsson; J. Sundberg; E. Jansson; W. Lassen Jordan; U. Rosenberg; and H. Sjogren, "Music Room Acoustics," *the Royal Institute of Technology in Stockholm by the Royal Swedish Academy of Music the Center for Human Technology, and the Center for Speech Communication Research and Musical Acoustics in April l975* , Vols. ISBN 91-85428-03-5 , no. the Royal Swedish Academy of Music 17 , p. 187, 1981.

[46] K. Merkelbach, "Feature Extraction for Musical Genre Classification," July 10, 2015.

[47] S. T. Pope, F. Holm and a. A. Kouznetsov, "Feature Extraction and Database Design for Music Software," in *Proc. 2004 Int'l Computer Music Conference* , Miami, 2004.

[48] G. Peeters, "A Large Set of Audio Features for Sound Description (Similarity and Classification) in the CUIDADO Projec," vol. version: 1.0(23 april 2004), p. 25, 2004.

[49] Sonal P.Sumare; and D.G.Bhalke2, "Automatic Mood Detection of Music Audio Signals: An Overview," *IOSR Journal of Electronics and Communication Engineering (IOSR-JECE)* , Vols. e-ISSN: 2278-2834,p- ISSN: 2278-8735, pp. 83-87, 2015.

[50] V. Hampiholi, "A method for Music Classification based on Perceived Mood Detection for Indian Bollywood Music," *World Academy of Science, Engineering and Technology International Journal of Computer, Electrical, Automation, Control and Information Engineering,* Vols. Vol:6, No:12, no. International Scholarly and Scientific Research & Innovation , pp. 1636 - 1643, 2012 .

[51] Bruno Rocha; Renato Panda; and Rui Pedro Paiva, "Music Emotion Recognition: The Importance of Melodic Features," in *In MML 2013: International Workshop on Machine Learning and Music* , Prague, 2013.

[52] Olivier Lartillot; Petri Toiviainen; and Tuomas Eerola, "A MATLAB Toolbox for Musical Information Retrieval," in *Data Analysis, Machine Learning and Applications, Studies in Classification, Data Analysis, and Knowledge Organization*, Finland, 2008.

[53] Murugappan Murugappan; Subbulakshmi Murugappan; and Bong Siao Zheng, "Frequency Band Analysis of Electrocardiogram (ECG) Signals for Human Emotional State Classification Using Discrete Wavelet Transform (DWT)," *J. Phys. Ther. Sci.,* Vols. Vol. 25, No. 7, pp. 753-759, 2013.

[54] Dadhi Dharma Wibawa; Mauridhi H Purnomo; Akhmad Marzuki; and LantanaD Rumpa, "Physiological Pattern of Human State Emotion Based on Ecg and Pulse Sensor," *Journal of Theoretical and Applied Information Technology,* vol. Vol.93.

No.1, no. © 2005 - 2016 JATIT & LLS. All rights reserved., pp. 81-87, 15th November 2016.

[55] Atefeh Goshvarpour, Ataollah Abbasi*, Ateke Goshvarpour, "An Accurate Emotion Recognition System Using ECG and GSR Signals and Matching Pursuit Method," *Biomedical Journal,* vol. 40, pp. 3 5 5 -3 6 8, 2017.

[56] Hany Ferdinando; Liang Ye; Tapio Seppänen; and Esko Alasaarela, "Emotion Recognition by Heart Rate Variability," *Australian Journal of Basic and Applied Sciences,* vol. 8(14) Special 2014, pp. 50-55, 2014.

[57] Marıa Teresa Valderas; Juan Bolea; Pablo Laguna; Montserrat Vallverdu; and Raquel Bailon, "Human Emotion Recognition Using Heart Rate Variability Analysis with Spectral Bands Based on Respiration," IEEE, 2015.

[58] Kohzoh Yoshin and Katsunori Matsuoka, *Correlation Between Mood and Heart Rate Variability Indices During Daily Life*, Ibaraki, Japan: SciRes, 2011..

[59] Robert LiKamWa; Yunxin Liu; Nicholas D. Lane; Lin Zhong, *MoodScope: Building a Mood Sensor from Smartphone Usage Patterns*, Taipei, Taiwan: ACM, 2013.

[60] H. David, "Perceptual and Cognitive Applications in Music Information Retrieval," Cognitive and Systematic Musicology Laboratory School of Music, Ohio State University, 2000.

[61] Alicja Wieczorkowska; Piotr Synakl and Zbigniew W. Ras, "Multi-label Classification of Emotions in Music," Polish-Japanese Institute of Information Technology, Koszykowa 86, 02-008 Warsaw, Poland; University of North Carolina, Charlotte, Computer Science Dept., 9201University City Blvd., Charlotte, NC 28223, USA; , Polish Academy of Sciences, Institute of Computer Science, Ordona 21, 01-237 Warsaw, Poland, 2006.

[62] Fang-Fei Kuo; Meng-Fen Chiang; Man-Kwan Shan; and Suh-Yin Lee, "Emotion-based Music Recommendation By Association Discovery from Film Music," ACM, Singapore., 2005.

[63] Van-Doan Nguyen; and Van-Nam Huynh, "On Information Fusion in Recommender Systems Based-on Dempster-Shafer Theory," in *IEEE 28th International Conference on Tools with Artificial Intelligence*, 1-1 Asahidai, Nomi, Ishikawa 923-1292, Japan, 2016.

[64] Yan Guo; Chengxin Yin; Mingfu Li; Xiaoting Ren; and Ping Liu, "Mobile e-Commerce Recommendation System Based on Multi-Source Information Fusion for Sustainable e-Business," MDPI, 2018.

[65] L. A. Zadeh, "A Simple View of the Dempster-Shafer Theory of Evidence and its Implication for the Rule of Combination," *THE AI MAGAZINE*, vol. California 94720, no. Research sponsored by the NASA Grant NCC-2-275, NESC Contract N00039-84-C-0243, and NSF Grant IST-8420416, pp. 85-90, 1986.

[66] K. Assefa, "The Significance of St. Yared’s Music in the Age of Globalization," in *Proceedings of the 16th International Conference of Ethiopian Studies, ed. by Svein Ege, Harald Aspen, Birhanu Teferra and Shiferaw Bekele, Trondheim* , 2009.

[67] Z. Bekele, "A Preview of Ethiopian Music," mediaethiopia.com, [Online]. Available: http://www.ethiopians.com/eth_musika.htm. [Accessed 18 4 2017].

[68] Upendra Shardanand; and Pattie Maes, "Social Information Retrieval: Algorithms for Automating "Word of Mouth"," ACM, Denver, Colorado, USA, 1995.

[69] Conor Hayes; and Pádraig Cunningham, "Smart Radio - Building Music Radio On the Fly," *Applications and Innovations in Intelligent Systems* , vol. VIII , no. Springer-Verlag London Limited 2001, pp. 129-138, 2001.

[70] Urszula Kuzelewskal; and Rafał Ducki, "Collaborative Filtering Recommender Systems in Music Recommendation," *ADVANCES IN COMPUTER SCIENCE RESEARCH*, vol. 10, pp. 67-79, 2013.

[71] Pasquale Lops; Marco de Gemmis; and Giovanni Semeraro, "Content-based Recommender Systems: State of the Art and Trends," in *Recommender Systems Handbook*, Bari, Italy, Springer Science+Business Media, LLC, 2011, pp. 73-105.

[72] Keiichiro Hoashi; Kazunori Matsumoto; and Naomi Inoue, "Personalization of User Profiles for Content-based Music Retrieval Based on Relevance Feedback," ACM, Berkeley, California, USA, 2003.

[73] Dmitry Bogdanov; Martín Haro; Ferdinand Fuhrmann; Anna Xambó; Emilia Gómez; and Perfecto Herrera, "Semantic Audio Content-Based Music Recommendation and Visualization Based On User Preference Examples," *Information Processing and Management*, vol. 49, no. Elsevier Ltd, pp. 13-33, 2012.

[74] Marius Kaminskas; and Francesco Ricci, "Location-Adapted Music Recommendation Using Tags," Springer-Verlag, Berlin, Heidelberg, 2011.

[75] Ivana Andjelkovic; Denis Parra; and John O'Donovan, "Moodplay: Interactive Mood-based Music Discovery and Recommendation," ACM, Halifax, NS, Canada, 2016.

[76] Aiym Sagdoldanova; Lyazzat Atymtayeva; and Zhanerke Yespolayeva, "Medicine Recommendation Technique by Using Dempster-Shafer Theory," *Advanced Engineering Technology and Application An International Journal*, Vols. 6, No. 3, no. Natural Sciences Publishing Cor., pp. 27-32, 2017.

[77] J. Donaldson, "A Hybrid Social-Acoustic Recommendation System for Popular Music," ACM, Minneapolis, Minnesota, USA, 2007.

[78] Lun-Wei Ku; and Cheng-Wei Sun, "Calculating Emotional Score of Words for User Emotion Detection in Messenger Logs," IEEE IRI , Las Vegas, Nevada, USA, 2012.

[79] Tuomas Eerola; and Jonna K. Vuoskoski, "A Comparison of the Discrete and Dimensional Models of Emotion in Music," *Sempre: Societ for Education, Music, and Psychology Research*, vol. 39(1), no. Reprints and permission: sagepub, p. 18–49, 2011.

Annexes

Annex A: Sample of Offline Survey Questionnaire

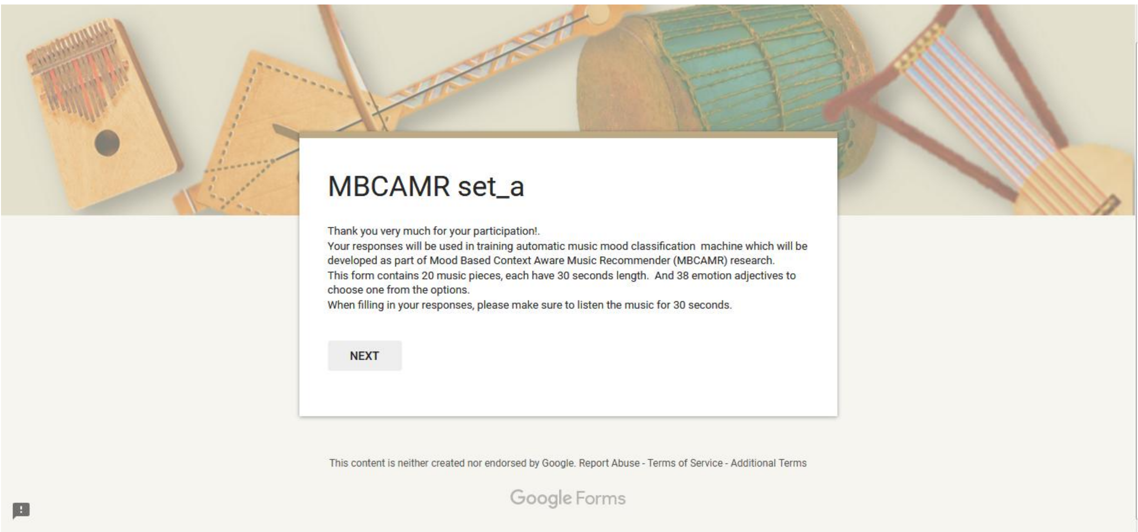
Dear friends
It would be very helpful if you can participate in the following study (and maybe forward to your friends and encourage them to participate).
Your responses will be used in training automatic music mood classification machine which will be developed as part of Mood Based Context Aware Music Recommender (MBCAMR) research.
This form contains 20 music pieces, each have 30 seconds length. And 38 emotion adjectives to choose one from the options.
When filling in your responses, please make sure to listen the music for 30 seconds
Thank you very much for your time

Thank you very much for your participation!.
Best Regards!!!
Addis Ababa University School of Graduate Studies Department of Computer Science

Gender:		Age:	
Artist Name	**Song Title**	**Audio Track**	**Mood**
Alemayehu Eshete	yewoyin aregitu	የወይን አረጊቱ	
Aster Awoke	fikir	ፍቅር	
Bizuayew Demissie	ende wof	እንደ ወፍ	
Dereje Degefa	mehed mehed alegn	መሄድ መሄድ አለኝ	
Ephirem Tamiru	kelelaye	ከለላዬ	
Geremew Assefa	bey endashash	በይ እንዳሻሽ	
Gossaye Tesfaye	chaw chaw	ቻዎ ቻዎ	
Hamelmal Abate	wedihalew	እወድሃለው	
Kibret Belay	abet afetater	አቤት አፈጣጠር	
Kuku Sebsibe	kolel	ኮለል	
Melkamu Tebeje	hawassa langano	ሐዋሳ ላንጋኖ	
Mikiyas Cherinet	girma modgesse nes	ግርማ ሞገሴ ነሽ	
Neway Debebe	yayine abeba	ያይኔ አበባ	
Tamirat Desta	kanchi ayibelitim	ካንቺ አይበልጥም	
Tewodros Kassahun	belsitegn	በል ስጠኝ	
Tewodros Kassahun	shee mendafir	ሼመንዳፍር	
Tigist Fantahun	ewedishalew belegr	እወድሻለው በለኝ	
Tilahun Gesses	fikirish new yegodag	ፍቅርሽ ነው የጎዳኝ	
Tilahun Gesses	yezenbaba mar nesh	የዘንባባ ማር ነሽ	
Zeritu Kebede	endaygelegn	እንዳይገለኝ	

Link to the audio file

Annex B: Sample of Online Survey Questionnaire

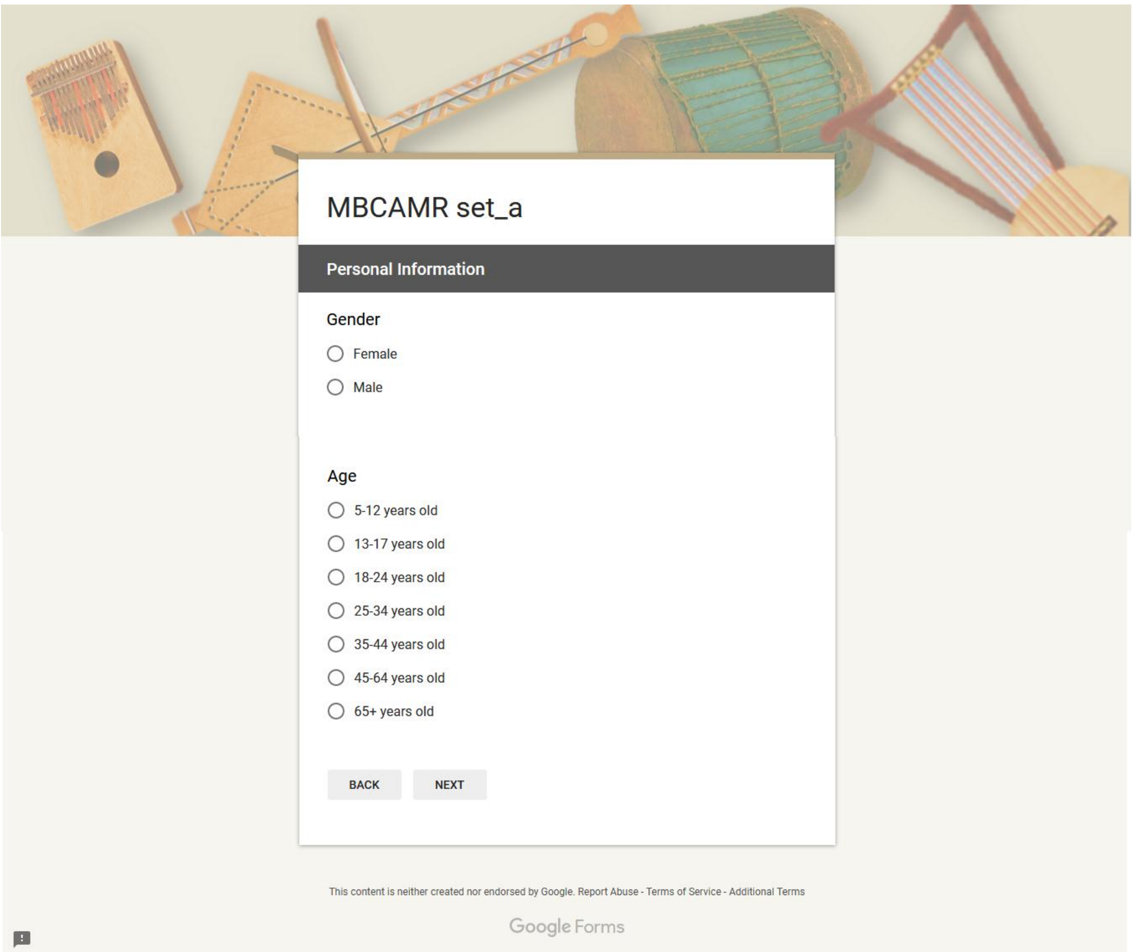

MBCAMR set_a

Thank you very much for your participation!.
Your responses will be used in training automatic music mood classification machine which will be developed as part of Mood Based Context Aware Music Recommender (MBCAMR) research.
This form contains 20 music pieces, each have 30 seconds length. And 38 emotion adjectives to choose one from the options.
When filling in your responses, please make sure to listen the music for 30 seconds.

NEXT

This content is neither created nor endorsed by Google. Report Abuse - Terms of Service - Additional Terms

Google Forms


MBCAMR set_a

Personal Information

Gender

- Female
- Male

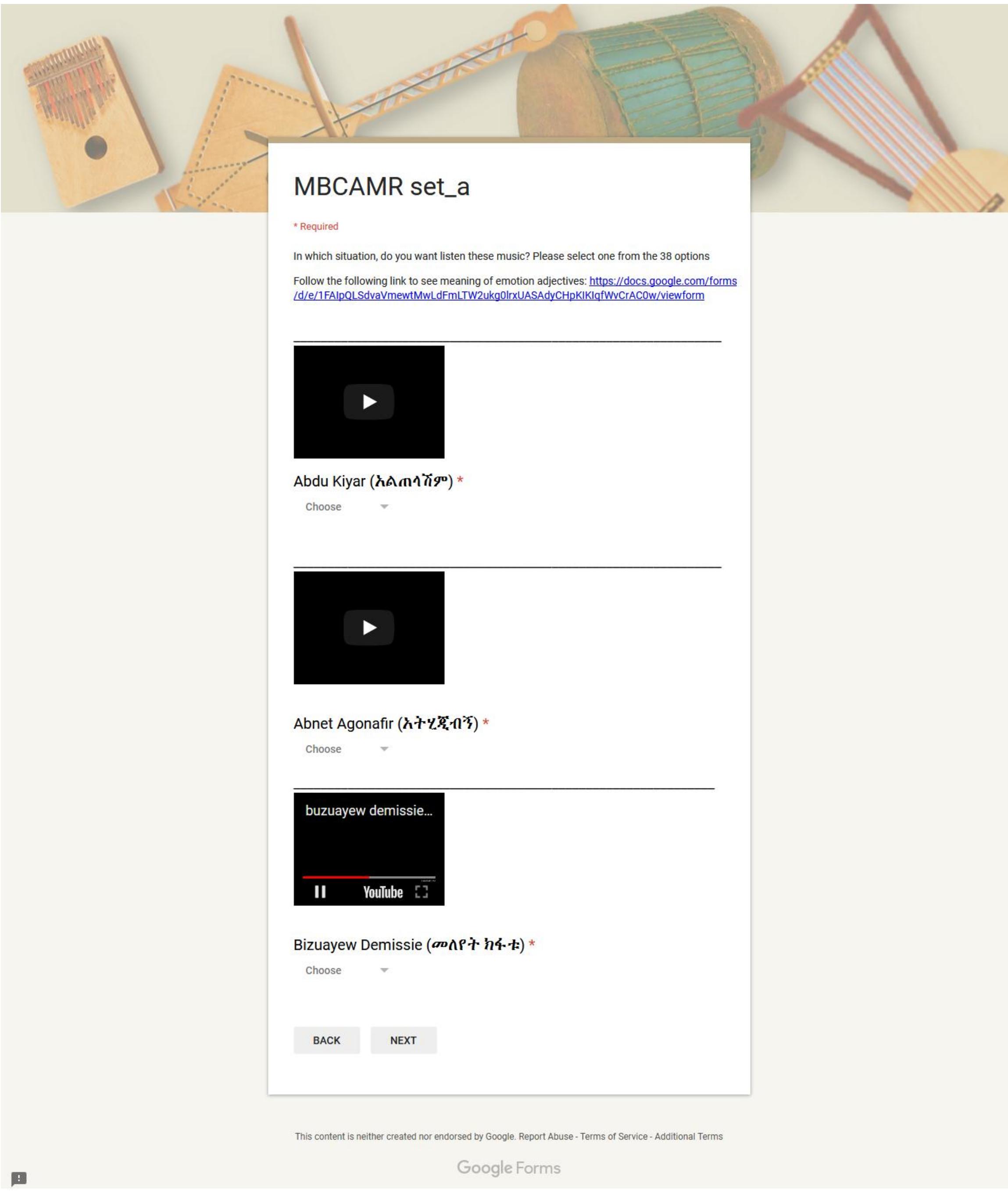
Age

- 5-12 years old
- 13-17 years old
- 18-24 years old
- 25-34 years old
- 35-44 years old
- 45-64 years old
- 65+ years old

BACK NEXT

This content is neither created nor endorsed by Google. Report Abuse - Terms of Service - Additional Terms

Google Forms



MBCAMR set_a

* Required

In which situation, do you want listen these music? Please select one from the 38 options

Follow the following link to see meaning of emotion adjectives: https://docs.google.com/forms/d/e/1FAIpQLSdvaVmewtMwLdFmLTW2ukg0lrxUASAdyCHpKIKIqfWvCrAC0w/viewform

Abdu Kiyar (አልጠላሽም) *

Choose

Abnet Agonafir (አትሂጂብኝ) *

Choose

buzuayew demissie...

YouTube

Bizuayew Demissie (መለየት ክፋቱ) *

Choose

BACK NEXT

This content is neither created nor endorsed by Google. Report Abuse - Terms of Service - Additional Terms

Google Forms

Annex C: All Features Extracted

Dynamics Elements

- RMS Energy Mean
- RMS Energy Standard deviation

Rhythmic Elements

- Fluctuation Mean
- Fluctuation Standard deviation
- Tempo
- Attack Time Mean
- Attack Time Standard deviation

Timberal Elements

- Zero Crossing Rate
- Low Energy Mean
- Low Energy Standard deviation
- Spectral Flux Mean
- Spectral Flux Standard deviation

Tonal Elements

- Chromagram Mean
- Chromagram Standard deviation
- Key Clarity Mean
- Key Clarity Standard deviation
- Key Strength Mean
- Key Strength Standard deviation
- Mode
- HCDF Mean
- HCDF Standard deviation

Spectral Elements

- Centroid Mean
- Centroid Standard deviation
- Brightness Mean
- Brightness Standard deviation
- Spread Mean
- Spread Standard deviation
- Skewness Mean
- Skewness Standard deviation
- Kurtosis Mean
- Kurtosis Standard deviation
- Roll off Mean
- Roll off Standard deviation
- Entropy of Spectrum Mean
- Entropy of Spectrum Standard deviation
- Flatness Mean
- Flatness Standard deviation
- Roughness Mean
- Roughness Standard deviation
- Irregularity Mean
- Irregularity Standard deviation
- MFCC Mean
- MFCC Standard deviation

Annex D: Sample Music Mood Dataset with Selected Features

Audio File	RMS Energy Mean	RMS Energy Std	Fluctuation Mean	Attack Time Mean	Entropy of Spectrum Mean	Roughness Mean	Roughness Std	Key Clarity Mean	Mode Mean	HCDF Mean	Tempo	Mood
Song1	0.104646	0.034138	168.1014	0.028353	0.702744	23.46491	16.79702	5.586207	-0.0415	0.372153	126.9858	LNA
Song2	0.12202	0.070358	212.9028	0.020825	0.760226	244.6597	334.9318	4.827586	-0.06092	0.349547	119.9379	HPA
Song3	0.066007	0.031884	194.9095	0.021773	0.821891	143.1744	170.5812	6.465517	-0.03218	0.403709	117.5683	P
Song4	0.088384	0.031819	140.9061	0.023722	0.747889	187.7168	169.7508	5.40678	-0.03553	0.314785	114.508	LNA
Song5	0.219816	0.088653	180.4514	0.023119	0.716832	1041.144	1197.139	5.271186	-0.04008	0.351675	92.22791	HPA
Song7	0.258049	0.057788	130.4331	0.013796	0.743018	1994.553	1248.74	8.258621	-0.09143	0.377798	120.5678	HPA
Song8	0.044483	0.021265	189.7302	0.023189	0.704343	55.78086	103.6553	6.40678	-0.02665	0.412076	70.90059	UnP
Song9	0.217954	0.067837	225.7064	0.027531	0.76465	1606.609	1898.055	7.62069	-0.15125	0.35156	172.2399	P
Song10	0.068198	0.02871	161.4441	0.027795	0.714387	99.79762	104.5778	8.016949	-0.01381	0.385036	99.99295	HPA
Song11	0.152694	0.064981	252.8331	0.024547	0.781983	1271.396	2477.544	8.050847	0.004094	0.371238	130.0158	SE
Song12	0.219166	0.088732	182.3185	0.02331	0.70529	911.8066	972.3761	7.741379	0.02347	0.360451	96.00833	P
Song13	0.034736	0.036756	229.0146	0.041418	0.66304	21.35316	54.92172	7.067797	-0.05383	0.383543	82.76315	UnP
Song14	0.015762	0.012097	135.0796	0.27556	0.771499	13.37408	23.9539	6.87931	0.05492	0.382979	125.3928	SE
Song15	0.079192	0.039359	149.4472	0.032408	0.703604	168.6936	191.813	6.40678	-0.06113	0.31281	67.8183	LNA
Song16	0.127569	0.054492	181.734	0.031537	0.645914	409.0645	477.497	5.603448	-0.04883	0.383243	85.10032	UnP
Song17	0.193045	0.096385	212.2315	0.021212	0.756439	770.3861	995.5517	7.559322	-0.02186	0.323614	61.51015	HPA
Song18	0.156139	0.065393	222.878	0.022885	0.686341	554.0922	463.1092	7.189655	-0.07169	0.355908	93.46212	LNA
Song20	0.171381	0.07462	188.979	0.02047	0.743895	694.1329	762.2506	7.827586	-0.06234	0.331308	115.8328	HPA
…	…	…	…	…	…	…	…	…	…	…	…	…

Annex E: Sample Source Code

Music Feature Extraction and Music Mood Detection Code Snippet

```
…
function pushbutton1_Callback(hObject, eventdata, handles)
…
…
set(handles.filepath, 'String', ' Please wait while system is working ...');
[filename pathname] = uigetfile({'*.wav'}, 'Select Music');
fullpathname = strcat(pathname, filename);
songpath = strcat('Selected Song >>  ', fullpathname);
set(handles.filepath, 'String', songpath);

a = miraudio(fullpathname);

%--------------------rms energy------------------------------
rmsenergy = mirrms(a, 'Frame');
RMSenergy_Mean = mean(mirgetdata(rmsenergy));
RMSenergy_meanS = num2str(RMSenergy_Mean);
set(handles.rms_mean, 'String', RMSenergy_meanS);

RMSenergy_Std = std(mirgetdata(rmsenergy));
RMSenergy_stdS = num2str(RMSenergy_Std);
set(handles.rms_std, 'String', RMSenergy_stdS);

%------------------- Fluctuation------------------------------
fluc = mirfluctuation(a);
Fluctuation_Mean = mean(mean(mirgetdata(fluc)));
Fluctuation_meanS = num2str(Fluctuation_Mean);
set(handles.fluc, 'String', Fluctuation_meanS);

…
…
…

%------------------- HCDF ---------------------
hcdf = mirhcdf(a);
HCDF_Mean = mean(mirgetdata(hcdf));
HCDF_meanS = num2str(HCDF_Mean);
set(handles.hcdf, 'String', HCDF_meanS);

%------------------- Mode ---------------------
mode = mirmode(a);
Mode_Mean = mirgetdata(mode);
Mode_meanS = num2str(Mode_Mean);
set(handles.mode, 'String', Mode_meanS);

%------------------- Key Clarity ---------------------
kc = mirkey(a,'Frame');
Key_Clarity_Mean = mean(mirgetdata(kc));
Key_clarity_meanS = num2str(Key_Clarity_Mean);
set(handles.key, 'String', Key_clarity_meanS);

%------------------- Tempo ---------------------------
tempo = mirtempo(a);
Tempo = mirgetdata(tempo);
tempoS = num2str(Tempo);
```

```
set(handles.tempo, 'String', tempoS);

%--------------------Classification--------------------
selectedFeatures = table([RMSenergy_Mean], [RMSenergy_Std],
[Fluctuation_Mean], [EntropyofSpectrum_Mean], [Key_Clarity_Mean],
[Mode_Mean]);
[label_hl, score_hl] = HLScore.predictFcn(selectedFeatures);

selectedFeatures = table([RMSenergy_Mean], [RMSenergy_Std],
[Fluctuation_Mean], [AttackTime_Mean], [EntropyofSpectrum_Mean],
[Roughness_Mean],[Roughness_Std], [Key_Clarity_Mean], [HCDF_Mean], [Tempo]);
[label_hp, score_hp] = HPScore.predictFcn(selectedFeatures);

…
…
…
selectedFeatures = table([RMSenergy_Mean], [RMSenergy_Std],
[Fluctuation_Mean], [AttackTime_Mean], [EntropyofSpectrum_Mean],
[Key_Clarity_Mean]);
[label_pu, score_pu] = PUScore.predictFcn(selectedFeatures);

selectedFeatures = table([RMSenergy_Mean], [RMSenergy_Std],
[EntropyofSpectrum_Mean]);
[label_su, score_su] = SUScore.predictFcn(selectedFeatures);

%----------------voting---------------------------
h_vote = 0; l_vote = 0; p_vote = 0; u_vote = 0; s_vote = 0;

if score_hl(2) < 0
    h_vote = h_vote + 1;
else
    l_vote = l_vote + 1;
end

if score_hp(2) < 0
    h_vote = h_vote + 1;
else
    p_vote = p_vote + 1;
end

if score_hs(2) < 0
    h_vote = h_vote + 1;
else
    s_vote = s_vote + 1;
end

…
…
…

if score_su(2) < 0
    s_vote = s_vote + 1;
else
    u_vote = u_vote + 1;
end

h_vote = h_vote * 0.84; l_vote = l_vote * 0.71;  p_vote = p_vote * 0.43;
s_vote = s_vote * 0.54; u_vote = u_vote * 0.72;
```

```
maximu = max([h_vote, l_vote, p_vote, s_vote, u_vote])
%------------------- Detecting Music Mood ---------------------
if maximu == h_vote
set(handles.resultMoodP, 'String', 'High Positive Affect');
set(handles.resultMoodS, 'Active', 'Elated', 'Enthusiastic', 'Excited',
'Peppy', 'Strong');
end

if maximu == l_vote
set(handles.resultMoodP, 'String', 'Low Negative Affect');
set(handles.resultMoodS, 'At rest', 'Calm', 'Placid', 'Relaxed');
end

if maximu == p_vote
set(handles.resultMoodP, 'String', 'Pleasantness');
set(handles.resultMoodS, 'Content', 'Happy', 'Kindly', 'Pleased',
'Satisfied', 'Warmhearted');
end
if maximu == s_vote
set(handles.resultMoodP, 'String', 'Strong Engagement');
set(handles.resultMoodS, 'Aroused', 'Astonished', 'Surprised');
end
if maximu == u_vote
set(handles.resultMoodP, 'String', 'UnPleasantness');
set(handles.resultMoodS, 'Blue', 'Grouchy', 'Lonely', 'Sad', 'Sorry',
'Unhappy');
end
```

Context Manager Code Snippet

Belief from Heart-rate

```
…

 def credPartion(heartrate):
    Pmax = 100.0
    Pmin = 30.0
    difHPA = abs(1 - abs((73.24 - heartrate) / (Pmax - Pmin)))
    difLNA = abs(1 - abs((72.13 - heartrate) / (Pmax - Pmin)))
    difP = abs(1 - abs((71 - heartrate) / (Pmax - Pmin)))
    difSE = abs(1 - abs((40 - heartrate) / (Pmax - Pmin)))
    difUnP = abs(1 - abs((58 - heartrate) / (Pmax - Pmin)))
    hbelief = min(difHPA, difLNA, difP, difSE, difUnP)
    totalBelief = difHPA + difLNA + difP + difSE + difUnP + hbelief
    return difHPA, difLNA, difP, difSE, difUnP, hbelief, totalBelief

def beliefAssignment(credHPA, credLNA, credP, credSE, credUnP, credH, totalCred):
    mHPA = round(credHPA / totalCred, 4)
    mLNA = round(credLNA / totalCred, 4)
    mP = round(credP / totalCred, 4)
    mSE = round(credSE / totalCred, 4)
    mUnP = round(credUnP / totalCred, 4)
    mH = round(credH / totalCred, 4)
    return mHPA, mLNA, mP, mSE, mUnP, mH
```

Belief from Text

```
def sentenceData():
    import csv
    HPA_sentences = []
    with open('D:/musicRecommendation/textDataset/hpasentences.csv') as csvDataFile:
        csvHPA = csv.reader(csvDataFile)
        for row in csvHPA:
            HPA_sentences += row

    LNA_sentences = []
    with open('D:/musicRecommendation/textDataset/lnasentences.csv') as csvDataFile:
        csvLNA = csv.reader(csvDataFile)
        for row in csvLNA:
            LNA_sentences += row

    P_sentences = []
    with open('D:/musicRecommendation/textDataset/p_sentences.csv') as csvDataFile:
        csvP = csv.reader(csvDataFile)
        for row in csvP:
            P_sentences += row

    SE_sentences = []
    with open('D:/musicRecommendation/textDataset/se_sentences.csv') as csvDataFile:
        csvSE = csv.reader(csvDataFile)
        for row in csvSE:
            SE_sentences += row

    UnP_sentences = []
    with open('D:/musicRecommendation/textDataset/unp_sentences.csv') as csvDataFile:
        csvUnP = csv.reader(csvDataFile)
        for row in csvUnP:
            UnP_sentences += row

    all_sentences = HPA_sentences + LNA_sentences + P_sentences + SE_sentences + UnP_sentences
    return HPA_sentences, LNA_sentences, P_sentences, SE_sentences, UnP_sentences, all_sentences

def tokzizeSentences(HPA_sentences, LNA_sentences, P_sentences, SE_sentences, UnP_sentences, all_sentences):
    from nltk.tokenize import word_tokenize
    HPA_tokens = [[w.lower() for w in word_tokenize(text)] for text in HPA_sentences]
    LNA_tokens = [[w.lower() for w in word_tokenize(text)] for text in LNA_sentences]
    P_tokens = [[w.lower() for w in word_tokenize(text)] for text in P_sentences]
    SE_tokens = [[w.lower() for w in word_tokenize(text)] for text in SE_sentences]
    UnP_tokens = [[w.lower() for w in word_tokenize(text)] for text in UnP_sentences]
    generic_tokens = [[w.lower() for w in word_tokenize(text)] for text in all_sentences]
    return HPA_tokens, LNA_tokens, P_tokens, SE_tokens, UnP_tokens, generic_tokens

def tfidfModeling(HPA_tokens, LNA_tokens, P_tokens, SE_tokens, UnP_tokens, generic_tokens, sentence):
    from nltk.tokenize import word_tokenize
    import gensim
    hpa_dictionary = gensim.corpora.Dictionary(HPA_tokens)
    lna_dictionary = gensim.corpora.Dictionary(LNA_tokens)
   …

    hpa_corpus = [hpa_dictionary.doc2bow(HPA_tokens) for HPA_tokens in HPA_tokens]
    …
    hpa_tf_idf = gensim.models.TfidfModel(hpa_corpus)
    lna_tf_idf = gensim.models.TfidfModel(lna_corpus)
…
…
```

```
…
    generic_sims =
gensim.similarities.Similarity('D:/musicRecommendation/textDataset/',generic_tf_idf[generic_corpus],
                                    num_features=len(genericDict))

    # New sentence prediction:

    new_sentence = [w.lower() for w in word_tokenize(sentence)]

    new_sentence_bow_hpa = hpa_dictionary.doc2bow(new_sentence)
    …
    new_sentence_tf_idf_hpa = hpa_tf_idf[new_sentence_bow_hpa]
    …
    simHPA = sum(hpa_sims[new_sentence_tf_idf_hpa])
    …

    return simHPA, simLNA, simP, simSE, simUnP, ovrAllSim

def beliefAssignment(simHPA, simLNA, simP, simSE, simUnP, ovrAllSim):
    total_score = simHPA + simLNA + simP + simSE + simUnP + ovrAllSim
    mt_HPA = simHPA / total_score
    mt_LNA = simLNA / total_score
    mt_P = simP / total_score
    mt_SE = simSE / total_score
    mt_UnP = simUnP / total_score
    mt_h = ovrAllSim / total_score
    return mt_HPA, mt_LNA, mt_P, mt_SE, mt_UnP, mt_h
```

Belief Combination

```
…
from __future__ import print_function
from pyds import MassFunction
from itertools import product
import userMoodDetHR
import UserMoodDetText

HPA_sentences, LNA_sentences, P_sentences, SE_sentences, UnP_sentences, all_sentences =
UserMoodDetText.sentenceData()
HPA_tokens, LNA_tokens, P_tokens, SE_tokens, UnP_tokens, generic_tokens =
UserMoodDetText.tokzizeSentences(HPA_sentences, LNA_sentences, P_sentences, SE_sentences, UnP_sentences,
all_sentences)

heartrate = float(input("Heartrate: "))
sentence = input("Sentence:")

credHPA, credLNA, credP, credSE, credUnP, credH, totalCred = userMoodDetHR.credPartion(heartrate)
mh_HPA, mh_LNA, mh_P, mh_SE, mh_UnP, mh_H = userMoodDetHR.beliefAssignment(credHPA, credLNA,
credP, credSE, credUnP, credH, totalCred)
m1 = MassFunction([({'h', 'l', 'p', 's', 'u'}, mh_H), ({'h'}, mh_HPA), ({'l'}, mh_LNA), ({'p'}, mh_P), ({'s'}, mh_SE),
({'u'}, mh_UnP)])

simHPA, simLNA, simP, simSE, simUnP, ovrAllSim = UserMoodDetText.tfidfModeling(HPA_tokens,
LNA_tokens, P_tokens, SE_tokens, UnP_tokens, generic_tokens, sentence)
mt_HPA, mt_LNA, mt_P, mt_SE, mt_UnP, mt_h = UserMoodDetText.beliefAssignment(simHPA, simLNA, simP,
simSE, simUnP, ovrAllSim)
```

```
m2 = MassFunction([({'h'}, mt_HPA), ({'l'}, mt_LNA), ({'h', 'l', 'p', 's', 'u'}, mt_h), ({'p'}, mt_P), ({'s'}, mt_SE),
({'u'}, mt_UnP)])

# **Dempster-Shafer Combination**
MassHR = m1.pignistic()
MassSent = m2.pignistic()
m3 = m1&m2
CombinedMass = m3.pignistic()

print("Mass of belief- Heartrate:")
print(" HPA:", round(MassHR.bel({'h'}), 4), " LNA:",round(MassHR.bel({'l'}), 4),  " P:",round(MassHR.bel({'p'}),
4),  " SE:",round(MassHR.bel({'s'}), 4), " UnP:",round(MassHR.bel({'u'}), 4))
print("\nMass of belief- Sentence:")
print(" HPA:", round(MassSent.bel({'h'}), 4), " LNA:",round(MassSent.bel({'l'}), 4),  "
P:",round(MassSent.bel({'p'}), 4),  " SE:",round(MassSent.bel({'s'}), 4), " UnP:",round(MassSent.bel({'u'}), 4))

print("\nCombined Mass of Belief:")
print(" HPA:", round(CombinedMass.bel({'h'}), 4), " LNA:",round(CombinedMass.bel({'l'}), 4),  "
P:",round(CombinedMass.bel({'p'}), 4),  " SE:",round(CombinedMass.bel({'s'}), 4), "
UnP:",round(CombinedMass.bel({'u'}), 4))

userMood = max(CombinedMass.bel({'h'}), CombinedMass.bel({'l'}),  CombinedMass.bel({'p'}),
CombinedMass.bel({'s'}), CombinedMass.bel({'u'}))

if userMood == CombinedMass.bel({'h'}):
    print("user Mood: HPA")

elif userMood == CombinedMass.bel({'l'}):
    print("user Mood: LNA")

elif userMood == CombinedMass.bel({'p'}):
    print("user Mood: P")

elif userMood == CombinedMass.bel({'s'}):
    print("user Mood: SE")

elif userMood == CombinedMass.bel({'u'}):
    print("user Mood: UnP")
```

Annex D: Mean (μ), Standard deviation (σ) and Beta (β) values of the selected features in each pair of music mood

		RMS Energy (Mean)	RMS Energy (Std)	Fluctuation (Mean)	Attack Time (Mean)	Entropy of Spectrum (Mean)	Roughness (Mean)	Roughness (Std)	Key Clarity (Mean)	Mode (Mean)	HCDF (Mean)	Tempo
HPA vs. LNA	μ	0.129312	0.057311	197.8651	…	0.732883	…	…	6.578804	-0.02716	…	…
	σ	0.067357	0.030478	38.6343	…	0.050022	…	…	1.354016	0.043359	…	…
	β	-0.8513	0.7616	-0.4057	…	-1.0518	…	…	0.2558	-0.2458	…	…
HPA vs. P	μ	0.1393	0.0612	207.7787	0.0258	0.7402	632.1545	771.3694	6.6667	…	0.363	122.4329
	σ	0.0682	0.03033	39.7321	0.0095	0.0498	652.2239	873.6262	1.3804	…	0.0409	28.8282
	β	-0.1003	0.164938	0.784716	0.086502	-0.58649	0.94116	-0.76409	0.305437	…	-0.3858	0.13365
HPA vs. SE	μ	0.142877	0.06523	…	…	0.753626	689.4049	933.038	…	…	0.355309	…
	σ	0.065185	0.030769	…	…	0.047205	715.3082	1188.459	…	…	0.044484	…
	β	-0.35712	0.328154	…	…	0.166214	0.401699	0.037207	…	…	-0.46882	…
HPA vs. UnP	μ	0.131942	0.056064	194.4385	0.029082	0.726032	…	697.3334	6.62411	…	…	…
	σ	0.084641	0.028624	41.46916	0.013408	0.053348	…	888.2416	1.406119	…	…	…
	β	-0.04352	-0.11838	-0.437	0.67547	-1.12018	…	0.274055	0.081945	…	…	…
LNA vs. P	μ	0.13128	0.058439	203.874	0.026909	0.723376	…	…	6.704188	0.359813	…	123.259
	σ	0.073012	0.033051	40.43739	0.008259	0.049223	…	…	1.26564	0.040371	…	31.54612
	β	0.984765	-0.77003	1.172413	-0.25696	0.711016	…	…	0.145339	-0.31627	…	0.004192
LNA vs. SE	μ	0.134947	0.062893	…	…	0.737891	620.2056	…	…	-0.02509	0.351057	123.7993
	σ	0.070449	0.03395	…	…	0.05125	702.7529	…	…	0.042893	0.043712	31.11622
	β	0.087182	0.425387	…	…	1.549646	0.275538	…	…	0.665046	-0.88314	-0.08716
LNA vs. UnP	μ	0.124604	0.053311	…	0.030295	0.710885	540.3033	599.8372	…	-0.02669	…	…
	σ	0.088675	0.030385	…	0.012757	0.049226	871.0696	743.7118	…	0.043349	…	…
	β	0.289423	-0.50372	…	0.416836	-0.45566	0.045552	0.322171	…	0.167328	…	…
P vs. SE	μ	0.150576	0.069698	216.5764	0.749437	760.8852	1022.655	…	…	…	0.347139	…
	σ	0.070412	0.033116	44.47763	0.049605	704.9699	1201.858	…	…	…	0.043412	…
	β	-0.65299	1.198623	-0.49841	1.320118	-0.51973	0.589728	…	…	…	-0.42148	…
P vs. UnP	μ	0.135009	0.056952	199.2626	0.029737	0.715008	…	…	6.749938	…	…	…
	σ	0.093164	0.03077	44.49743	0.013584	0.052023	…	…	1.334245	…	…	…
	β	0.004313	-0.04575	-0.92106	0.703663	-0.57706	…	…	-0.20437	…	…	…
SE vs. UnP	μ	0.138223	0.060994	…	…	0.728535	…	…	…	…	…	…
	σ	0.091745	0.031861	…	…	0.056117	…	…	…	…	…	…
	β	0.177916	-0.33317	…	…	-0.83366	…	…	…	…	…	…

Declaration

I, the undersigned, declare that this thesis is my original work and has not been presented for a degree in any other university, and that all source of materials used for the thesis have been duly acknowledged

Declared by:

Name: ______________________________________

Signature: ___________________________________

Date: _______________________________________

Confirmed by advisor:

Name: ______________________________________

Signature: ___________________________________

Date: _______________________________________