Journal of Engineering and Applied Sciences 13 (Special Issue 8): 6378-6389, 2018
ISSN: 1816-949X

Highly Efficient and Robust Audio Identification and Analytics System to Secure Royalty Payments for Song Artists

Tharika Madurapperuma, Gothami Abayawickrama, Nesara Dissanayake,
V.B. Wijesuriya and K.l. Jayaratne
School of Computing, University of Colombo, No. 35 Reid Avenue, Colombo 7, Sri Lanka

Abstract: Radio is yet one of the most popular broadcast media in many emerging regions and operates as one of the principle sources of income for licensors and publishers, collectively termed as owners of copyrighted music. Royalty payments are legal obligations towards owners of copyrighted music under intellectual property rights legislations. Independent conjoint monitoring of copyrighted music being broadcast over every radio station on each day is an intricate requisite for upholding the said legislations of a country or region and therefore, requires automated techniques that are efficient, scalable and robust to both radio and content noise. We consider the development of an automated radio broadcast audio monitoring system with identification and analytics capabilities to assist collection of royalty payments from radio music licensees for copyrighted music, specifically for songs. First, we pre-process a broadcast radio stream to identify song segments, aka objects via. an efficient onset detection mechanism. Next, we perform efficient hashing of identified objects in the frequency domain and compare generated hashes with those in the entries of a pre-compiled copyrighted song database using an efficient hash matching technique. Subsequently, our system stores broadcast data of the identified music objects (e.g., timestamp, name of lyricist, etc.) in the database for later use from an analytic application.

Key words: Content based audio identification, visual analytics, royalty payment, broadcast, mechanism, Colombo

INTRODUCTION

Radio has become an interestingly popular broadcast medium which dedicates a considerable amount of airtime to entertain masses via. distinct works of music. Most of these works are copyrighted material and therefore, airing of such work over a radio broadcast medium may require a legal commitment from a designated radio music licensee in terms of a royalty payment to its respective owner (s). Radio stations broadcast hundreds of songs each day and currently thousands of radio stations operate within nations, regions and some globally. The need to monitor such large number of radio stations each with few gigabytes of aired content per day by an independent organization (s) to assist collection of royalty payments poses a number of interesting problems: how can one collect royalty payments for each individual work of music aired on each station? is manual monitoring and identification of every work a feasible solution? who ensures the credibility of the overall process? is there a way to minimize copyright infringements of such works?

We propose an automated solution to address the above stated problems and implement our solution using modern software engineering principles and development technologies. Even though we limit our narrative to copyrighted songs, we believe that our identification algorithm will work for any type of music work which is necessarily continuous, i.e., without significant number of silent points (explained later) in the audio stream of the song object which will other wise be classified as a non-song object. We believe that our system will benefit both licensors and publishers of copyrighted songs from here on collectively referred to as song artists. In some related literature they are also, referred to as rights holders (Carlson, 2011; Milosalvsky *et al*., 2010). Hence, we use these two terms interchangeably in this study. We refer to system owner (s) as the organization (s) responsible for carrying out independent conjoint monitoring of radio stations to uphold intellectual property rights legislations for copyrighted songs.

Corresponding Author: Tharika Madurapperuma, School of Computing, University of Colombo, No.35 Reid Avenue, Colombo 7, Sri Lanka

System capabilities: Our solution is a web based system which:

- Continuously monitor and identify all copyrighted songs broadcast through a given set of radio stations
- Store broadcast data of identified songs for later use (e.g., for generation of reports or analytics in general)
- Perform analytical reasoning via. visual interfaces on the stored data
- Store unidentified songs and associated broadcast data for manual resolution

The original intent of this system was to monitor all the radio stations in Sri Lanka. However, we note that it can be used for general conjoint audio monitoring regardless of the context. To facilitate identification of copyright songs, an up-to-date Mongo DB DataBase of songs released so far in Sri Lanka is compiled. Any song that is not in the database but broadcast through a station is designated as an unidentified song. As part of the system, we propose a novel content-based audio identification technique which is both highly efficient and robust. Our technique is content-based in the sense that it has the ability to filter out non-song segments or objects from the continuous radio broadcast stream to extract song objects to feed into the identification process.

We develop a web application to integrate different components needed for overall system functionality. It provides the interface to the up-to-date copyrighted song database, facilitates management of metadata of entries in the database corresponding to copyrighted songs, song artists and radio stations. The application also provides functionality to generate customizable payment and other types of reports for both system owners and song artists with support for visual analysis of relevant data.

Novelty of contribution: Pertaining to Sri Lankan context to the best of our knowledge, we are unaware of a comprehensive system other than that of ours for independent conjoint monitoring of copyrighted songs broadcast through radio stations in the country, hence, demonstrating the novelty in the domain of Sri Lankan music industry.

Visual analysis of the broadcast trends and patterns of musical works aired on radio, facilitates to make important, critical and financially valuable decisions. We believe that song artists and other interested entities would definitely prefer to gain access to such worthwhile information. Since, our system can be independently managed and all broadcast data are readily available for analysis, we also propose a novel business model based on visual analytics capabilities of our system, one could exploit for financial gain.

Our identification technique is also original in terms of the silent points based onset detection, efficient hash computation and matching techniques.

Related systems: Shazam (Wang, 2003) is a popular music identification application for PCs and smartphones. It uses the built-in microphone in a smartphone or a computer to record a short audio clip of a musical work. If the work is identified, Shazam outputs related details such as song title, singer, album, etc., to the user, claims that the work is unknown otherwise. Opposed to our system which continuously monitors all broadcasts of multiple radio stations, Shazam only identifies musical works of shorter time duration on per user basis. It generates audio fingerprintsfor the entirety of the audio clip and matches them against the fingerprints of musical works stored in their large database. An identification system based on audio fingerprinting which queries for possible entries in a large library of songs is also, proposed in (Haitsma and Kalker, 2002). The main difference between Shazam and our solution is that our system is designed and implemented for continuous, concurrent monitoring and identification of songs from comparably longer streams of broadcast audio.

A similar research by Senevirathna and Jayaratne (2013) also, describes a content based audio identification system for broadcast radio where a peak based feature extraction method in frequency domain is used to generate hash values for songs. The main drawback of this approach is that in order to generate a hash value, the difference in frequencies of twopeak pointsand also the time difference between them are taken into account which results in a considerable false positive rate. The reason behind this is that the same frequency and the respective time difference can intuitively be included in multiple songs. Furthermore when the matching process is over (Senevirathna and Jayaratne, 2013) selects the database entry with the most number of matches as the matching candidate entry for the broadcast song without considering the order of the hash matches over the timeline of the song object whereas we perform the matching taking into account the respective order of the hash matches over the timeline. As of Senevirathna and Joyaratne (2015) our identification approach is also, robust to modifications (added cliche, copyright musicincluded in a commercial, etc.) of or noise (thermal, electromagnetic, explicit, etc.) added to the original copyrighted song possibly by the radio station or any other entity before broadcasting. Also, our approach does

not require the entire song to be aired or broadcasted separately from the rest of the content for accurate identification. Moreover, the implementation of Senevirathna and Jayaratne (2013) is not highly scalable as that of ours.

An automated media and content reporting system for broadcast media which performs royalty payment calculations while tracking and analyzing broadcast content and compensating the appropriate rights holders the appropriate amount is described by Carlson (2011). This study recommends employing artificial intelligence whereas we employ a hash based efficient identification technique that scales well.

A patented system which focuses on the dynamic processing of royalties for song downloads is described by Milosalvsky *et al*. (2010). Here, the main focus lies on processing royalty payments based on existing information as soon as it becomes available. On the contrary our system not only enables automatic calculation of royalty payments but also, identifies copyrighted music from live broadcast media. However, our system does not cater to analyzing electronic songs downloads over the internet as (Milosalvsky *et al*., 2010) anticipates because our focus is only in radio broadcast monitoring.

ACR cloud broadcast monitoring services (Anonymous, 2016) is an automatic content recognition platform that allows users to upload their own media content and ingest a live stream for content identification and monitoring. This system provides the capability to perform data analysis so as to acquire competitor insight and predict trends by monitoring and analyzing live streams. While we specialize and focus on providing broadcast monitoring of songs through radio, (Anonymous, 2016a-c) also, caters to the identification of other types of musical works such as advertisements, custom audio and video content which does not comply with the problem we try to address.

Overview of the implementation: A large number of new song albums and singles are released every year and most of these new releases are readily aired on broadcast radio to proliferate their popularity. There by the business requirements associated with automated monitoring of new releases for royalty collection, data analytics, etc. have become increasingly complex In modern music industry. Hence, factors such as performance, maintainability and flexibility are of paramount importance for the system we propose. In this section, we briefly describe important software engineering principles and IT infrastructure that we incorporate to implement our solution thereon to fulfill complex business requirements.

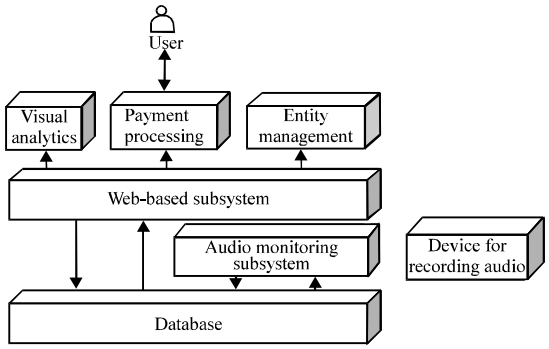

Fig. 1: System architecture of the proposed system. The arrows in the diagram show the direction of data flow between each layer

The system architecture is shown in Fig. 1. System users interact with the web-based subsystem through entity management, visual analytics and payment processing components. The visual analytics component retrieves data from the database through the web-based subsystem and renders the information visually to users. Similarly, the payment processing component retrieves data from the database and generate useful reports. The entity management component is concerned with managing details about songs, artists, radio stations and users of the system.

Audio monitoring subsystem concurrently monitors many radio stations with the help of an external hardware device, performs identification of copyrighted songs and interacts with the database for both retrieval and updating of information pertaining to the identification process.

We select the iterative and incremental software development framework Scrum (Lima *et al*., 2012) which falls under the Agile software development umbrella to develop our system. The adoption of the Scrum framework enables the use of software engineering techniques such as version controlling, continuous integration and continuous deployment during development.

Owing to scalability, availability, reliability, security, fault tolerance, cost effectiveness and processing power requirements, we employ cloud infrastructure for the development and deployment of our system. We note that using cloud infrastructure is not a necessity for development and deployment of our system, however, cloud infrastructure greatly simplifies the requirements pertaining to each stage. The elasticity and load-balancing capabilities offered through "Infrastructure as a Service (IaaS)", enable auto-scaling of resources to support high volume of user activity on the web-based subsystem. We use cloud infrastructure

services offered through Amazon Web Services (AWS) (Anonymous, 2016), especially, services such as Amazon Elastic Compute Cloud (EC2), AWS Elastic Beanstalk and AWS Code Pipeline are of great importance to us. The web-based subsystem is deployed through AWS Elastic Beanstalk service while the audio monitoring subsystem runs on Amazon EC2 instances. Assuming the number of radio stations and their daily program durations are static, ceteris paribus, the resource requirements for the component can easily be predetermined unlike for the web-based system which interacts with external users. Therefore, policies for capacity provisioning and auto-scaling are manually configured on the EC2 instances. The audio recording device is set-up on-site and recorded audio streams from each radio station are first multiplexed into one data stream and subsequently uploaded to a master EC2 instance where the stream is demultiplexed and each resulting audio stream is directed to the respective EC2 instance for further processing (encompassing identification).

The database for storing data relating to songs, broadcasts and other system functionalities is set-up in a separate EC2 instance with policies additionally configured for load-balancing. Mongo DB (Anonymous, 2016), a NoSQL document-oriented database program, has been used to implement the database. We leverage powerful capabilities of Mongo DB such as efficiency, scalability (esp. horizontal scalability), support for semi-structured immutable data (where data about songs is an appropriate realization of this) deep query-ability for data analytics and schema-less easy persistence of application objects. The distributed deployment architecture facilitates fault tolerance for each system component. For example, although, the instance containing the web application terminates unexpectedly, this does not affect the functionality of the audio monitoring subsystem or the database. Figure 2-4 show the entire development and the deployment set-up of the system.

MATERIALS AND METHODS

Audio monitoring: We present in this section, the main contribution of this study. We describe in several stages the functionalities of our novel audio monitoring approach which consists of filtering the broadcast stream for song objects and subsequently identifying the song being aired. The identification process finds a matching entry in the database by comparing features of the (song) audio stream at hand with those of songs in the database and records the details pertaining to the outcome of the matching process in the database.

Choosing the most corresponding audio identification methodology is important, since, that

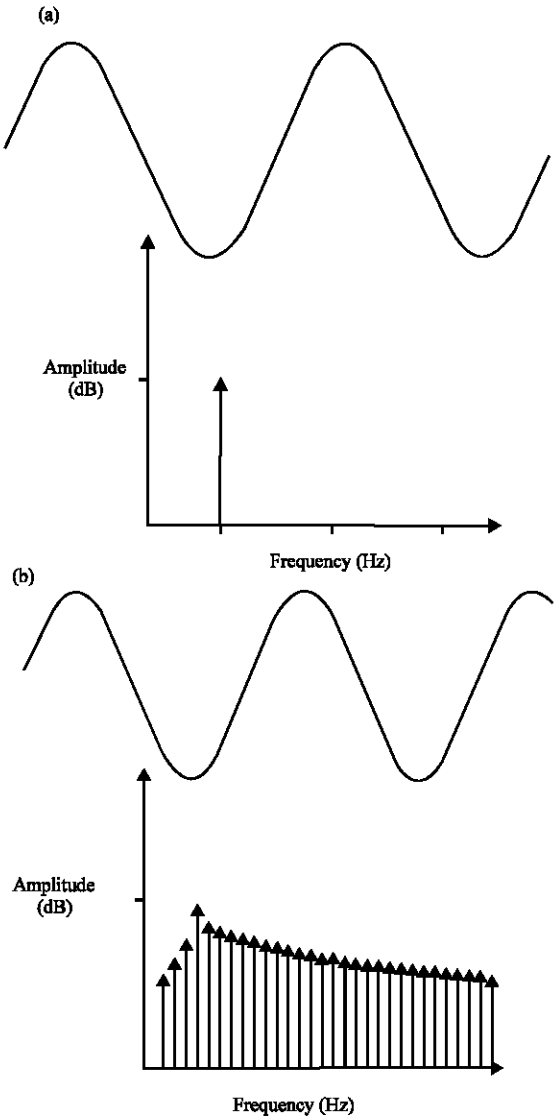

Fig. 2: a) Measuring an integer number of periods (top) gives an ideal FFT (bottom) and b) Measuring a non-integer number of periods (top) adds spectral leakage to the FFT (bottom)

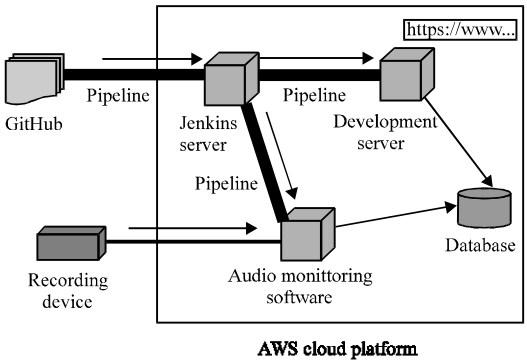

Fig. 3: System development and deployment set-up

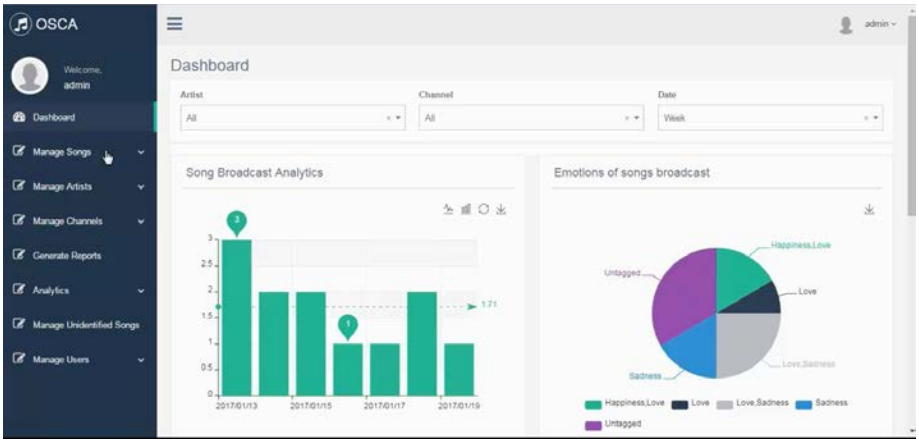

Fig. 4: An interface to visual analytics in the web-based subsystem

would affect the performance and the usage of the system. There are several approaches that can be followed to identify songs but when considering the identification of songs in radio broadcasts and royalty payments there are some special facts that need to be considered. The literature survey by Senevirathna (2015) indicates several such methodologies and usages of each method. By considering them, we found that one of the best methods for identification of songs in radio broadcasts is audio fingerprinting. Hence, we selected this approach.

Obtaining radio streams: A device capable of capturing radio broadcasts is used for obtaining the radio streams required for audio identification. A cron job is utilized for periodic execution of this process concurrently for each radio station in powerful cloud servers.

Pre-processing: We first adjust the sampling rate of the recorded radio broadcast stream from each radio station to 11025 Hz. The reason behind this is as follows. The hardware device records the streams in MP3 format and most commonly used sampling rate for MP3 is 44100 Hz. Our approach seeks to process 24 h long MP3 files (recorded broadcast streams) within 24 h and therefore we do not consider very high frequency values for the identification process. In fact, we disregard frequency values more than 5512.5 Hz. This threshold value is decided upon a reasonable upper bound on the frequencies of musical notes commonly appearing in songs (Anonymous, 2016a-d). Conversely, we assume that musical notes whose frequencies are above the specified threshold value do not commonly occur in song performances. According to Nyquist-Shannon sampling theorem, the sampling rate should be at least twice the maximum frequency of the signal, thus the sampling rate 11025 Hz (5512.5 Hz×2). Afterwards, the recorded radio streams are converted to Pulse-Code Modulation (PCM) data. This PCM data is kept in 512 sized buffers and to keep track on time values of data, each buffer is assigned with a time value which is incremented in order. There on the audio data is transformed from stereophonic to monophonic by averaging the two channels.

We apply Hamming window to monophonic PCM data. Our approach performs peak extraction in the frequency domain. FFT is applied by assuming that the data is finite with a continuous spectrum that is of length one period of a periodic signal (Fig. 2a). When the measured signal is periodic and an integer number of periods fill the acquisition time interval, the FFT turns out fine as it matches this assumption. However, most of the time, a signal does not contain an integer number of periods (Fig. 2b). Windowing can minimize the effect from non-integer number of cycles in a signal (Le Berre and Parrain, 2010). There are no best or worst windows out there. Each type of window is suitable for the problem it deals with. In our context, dealing with noise is an important problem and identification of peaks clearly amidst noise is necessary. Therefore, we use Hamming window, a window which is capable of handling noise, as the windowing function as it separates peaks clearly in this regard. Thus, maximum frequency values (or peaks) can be extracted easily.

Filtering non-song objects: A radio broadcast stream consists of many non-song objects such as advertisements, news, drama, talk shows, etc. Identifying non-song objects and removing them before the stream is processed for identification, makes the entire monitoring process more efficient. We achieve this filtering policy by

using a mechanism which detects silent points in the radio stream, there by identifying non-song objects and subsequently filtering them out. Silence-based filtering of non-song objects has also been studied by Senevirathna and Jayaratne (2013).

Loosely stated, a silent point refers to a point on the time series of a radio broadcast stream where the entropy or the energy of the unmodulated audio signal is closer to zero. Moreover, a silent point in a fourier transformed audio signal refers to points where the frequency is also closer to zero. Silent point is a relative measure that depends on the energy (or strength) of the audio signal. Therefore, one would need to define a threshold (or bound) on the energy in an appropriate measure, locally to the signal. Any point on the time series where the signal energy drops below the threshold, can be identified as a silent point. Compare Fig. 3a, b that show the wave patterns for a commercial and a news item respectively against Fig. 3c which shows the wave pattern of a song. The circles indicate silent points. Clearly, one can use the pattern (s) of the silent points (e.g., time between two silent points) to distinguish between a commercial or news item and a song.

In general, most of the non-song objects contain a set of frequent silent points though it is not the case in most of the song objects. We claim that this approach to filtering greatly increases the overall efficiency since, it is simple to compute and the identification process only needs to work with song objects if the filtering process succeeds. Even if the filtering process fails at some point (when songs themselves contain frequent silent points) it will not hinder the accuracy of the rest of the system components. A detailed description of the filtering process for non-song objects is provided in study.

A steps of filtering non-song objects from broadcast radio stream: The filtering approach contains a series of steps as follows. A threshold energy value is decided based on the average energy of the audio stream. Given a particular time frame, the number of silences are counted. A silence is determined based on the fact that the amplitude of the audio signal lies below the threshold energy value throughout the time frame considered. If more than ten silences were counted it was regarded as a non-song object and discarded. Several experiments were conducted by taking different thresholds based on different methods and different time frames.

Set of experiments are targeted at finding an average value for the energy. Samples of radio commercials, religious talks, radio drama, news, songs and talk shows were taken for the experiment to arrive at an average energy value. The stereo signal was converted into mono for as pre-processing step. In the next step, all the negative values are discarded to avoid nullifying the positives to negatives when calculating the average energy of the signal. Following a series of experiments thereby, it was decided to use 0.035 as the average energy of the signal. The signal was divided into frames of 0.1 sec which was chosen after a series of experiments conducted using different values.

According to the results of the evaluation carried out for 100 different audio objects, a song object could be distinguished from non-song objects with a 96% accuracy whilst the accuracy for distinguishing non-song objects from songs was 57%. This reduction for non-song objects is due to the fact that some commercials and drama consist of continuous background music of original songs. But these results were subjected to some restrictions as follows.

- Use of silent points may fail when advertisements contain background music of an actual song being played for a considerable amount of time
- When a song is played for a very short amount of time or if a song contains many silences in the middle with no singing
- A drama having lengthy music clips included

Development and deployment set-up on the cloud
An interface to visual analytics

Peak extraction and hash generation: Features extracted from the audio data need to be converted into proper representations that can be persisted in a database and used for subsequent computations. Therefore, using some important distinguishable features of audio data, set of hashes are generated per song object. Features are extracted from the frequency domain so that robust identification is made possible even if the audio signal is affected by considerable amount of noise or distortion. To convert the (time series) data into frequency domain, Fast Fourier Transformation (FFT) is used.

After converting audio data into frequency domain, frequencies at which the signal contains maximum energy are selected as values suitable for generating hashes: we call these as peak points. Moreover, these peak points are extracted by considering maximum amplitude value in a selected frequency range. We use a number of frequency ranges to make sure that the hash coverage for the song object is evenly spread across the entirety of its frequency spectrum. We extract a set of consecutive peaks over the frequency ranges for a buffered portion of the stream and generate a single hash value for 4 consecutive sets of peaks by concatenating all subsequent peak frequencies. The next hash value is

generated by further extracting a set of consecutive peaks and discarding the foremost set. In order to identify a song whose broadcast commences at any point after its original intended beginning which is a quite common scenario in radio broadcasts, a sliding window approach is used when selecting consecutive peaks.

For an example, let us say we have selected frequencies with maximum amplitude from six frequency ranges (one peak from one frequency range). Input for the hash function is a set of frequencies and output of the hash function is a string generated using a set of consecutive peak frequencies. Equation 1 and 2 shows sample calculations of two hash values:

Peak set 1 → f11, f12, f13, f14, f15, f16
Peak set 2 → f21, f22, f23, f24, f25, f26
Peak set 3 → f31, f32, f33, f34, f35, f36
Peak set 4 → f41, f42, f43, f44, f45, f46
Peak set 5 → f51, f52, f53, f54, f55, f56

Hash1 = peakSet 1: concat (peakSet2): concat (peakSet3): concat (peakSet4):toString() (1)

Hash 2 = peakSet2: concat (peakSet3): concat (peakSet4): concat (peakSet5) :to String () (2)

For each song object, a set of hash values are generated. It happens in two different circumstances. One is when inserting the original song into the database (also, called registration) and the other is when matching the radio stream with the song in the database. Both scenarios use the same hash generation methodology. The hash values from the radio stream are matched with a very large set of hash values of all songs in the database. Note that hash values of the original songs are stored in the database when they are registered by the system owners.

Matching: Since, hash value generation is based on frequencies, we attach a temporal value to a hash value to facilitate the matching of hash values over time. In the matching phase, hash values generated from the broadcast stream of a song are searched through the database. When matches are found, temporal values are used to find out whether the several hash matches occurred consecutively in time for those subsequent hash values of song registered in the system. More elaborately as shown in Fig. 4, hash matches should demonstrate a straight line in the plot if a broadcast song is correctly

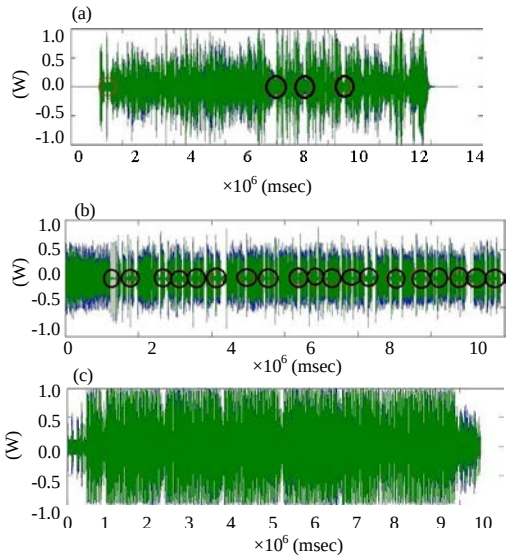

Fig. 5: a) Wave pattern for a commercial; b) Wave pattern for a news item and c) Wave pattern for a song

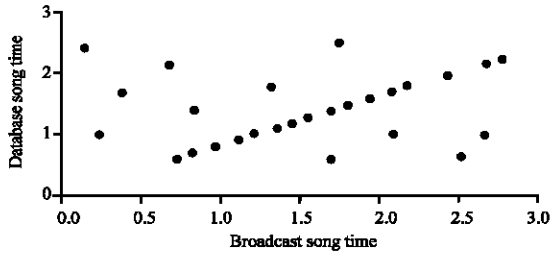

Fig. 6: Matching hash locations

matched with a registered song in the database: which is the case here. To mathematically validate this claim we use linear regression. According to the standard regression formula Y = bX+a for a song to be correctly matched, the value of the b should be closer to 1 since, the time difference between subsequent matching hash values for both broadcast song and original song (in database) should almost be the same. Therefore if the regression score is above a prespecified threshold regression value, the broadcast song can be confirmed as an Identified song. Pseudo code for the two algorithms used for song matching and registration respectively are shown in Fig. 5 and 6. (A comprehensive description of these algorithms are availabl eathttps://github.com/Gothami/Audio-Identification-Algorithm A comprehensive description of these algorithms are available athttps://github.com/Gothami/Audio-Identification-Algorithm (Algorithm 1 and 2). A high level flow diagramdescribing the audio monitoring process is shown in Fig. 7.

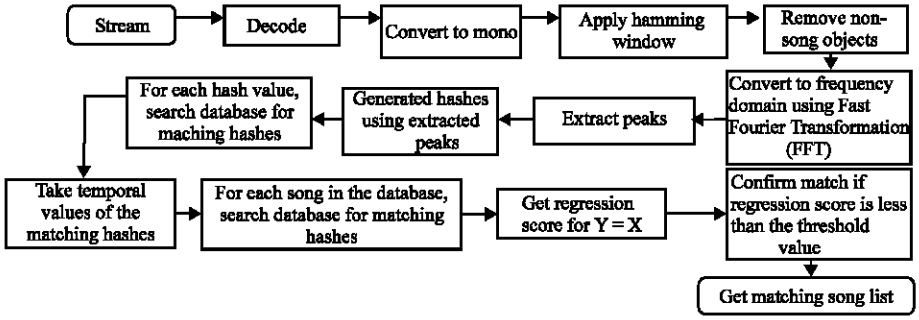

Fig. 7: High-level flow diagram of the audio monitoring and identification process

Algorithm; 1 Algorithm for song registration:

```
Inputs:
        \mp3File, songID
Initialize:
        int[] peaks, List<int[]> in time peakList, = 0, String
        hash, TreeMap<String, SongInfo> hashMap
while not isEnd(mp3File) do
buffer ← Decode (mp3File, 512)
        time + = 1
        buffer ← Monophonic (buffer)
        buffer ← Hamming
        realfft ← RealFFT (buffer)
        peaks ← GetPeaks (realfft)
        peakList.add (peaks)
        if sizeOf (peakList) = 4 then
        hash GenerateHash (peakList)
                hashMap.add (hash, neSongInfo (songID, time))
                peakList.remove (0)
        end if
end while
AddtoDB(hashMap)
```

Algorithm 2; Algorithm for song identification via. hash matching:

```
Inputs:
        stream
Initialize:
        int songTimes[SONGCOUNT], double
        regressionScores[SONGCOUNT], LinkedList<int[]>
        slidingWindow
 while not isEnd(stream) do
        buffer ← Decode(stream, 512)
        buffer ← Monophonic(buffer)
        buffer ← HammingWindow(buffer)
        realfft ← RealFFT(buffer)
        peaks ← GetPeaks(realfft)
        peakList.add(peaks)
        if sizeOf(peakList) = 4 then
          hash ← GenerateHash(peakList)
          peakList.remove(0)
          songInfoList ← QueryDB(hash)
          Reset(songTimes)
          if songInfoList is not Null then
                for each songInfo in songInfoList do
                    songTimes[songInfo.SongID]← songInfo.Time
                end for
        end if
        slidingWindow.Add(songTimes)
        if sizeOf(slidingWindow) = MAXSLIDINGWINDSIZE then
                slidingWindow.Remove(0)
        end if
        song ← 0
        while song < SONGCOUNT do
      matchesForSong←SelectMatchesForSong(slidingWindow,song)
                matchSet←PickSameGapMatches(matchesForSong)
                closestMatches ? PickClosetMatches(matchSet)
                if SizeOf(closestMatches) > 10 then
                        score←GetRegressionScore(closestMatches)
                        if score > regressionScores[song] then
                                regressionScores[song]← score
                        end if
                end if
                song += 1
        end while
    end if
end while
return regressionScores
```

Payment processing: The aims and objectives of the payment processing component include:

- Customizable payment processing and automated invoice generation for song artists, radio stations and system owners
- Development of a business rules engine capable of customizing payments related logic according to needs of the system owners

The system is capable of processing payments for system stakeholders. The system owners usually send payment related reports and invoices to song artists and radio stations. For verification purposes, the latter are allowed to generate the same information (pertaining to their domain of interest) through the system. Customizable reports on broadcast history of songs are also generated through the system.

Business rules generation: It may be the case that royalty payment related business logic need to be tailored to handle special cases. Consider the following scenarios:

- A change in the royalty tariff rates proposed by state legislation
- The original artist being awarded a partial sum of the intended royalty fee for a cover song or a remix
- Change of the intended person for royalty payments this can occur in situations where an artist passes away

A sample propositional formula for the third scenario above can be shown as follows:

$$(P \wedge R) \wedge Q \Rightarrow S$$

Where:

P = X is a song artis
Q = X has passed away
R = Y is eligible to become the rights holder in lieu of X
S = Royalty payment ownership transferred from X to Y

The associated business rules for specialized cases are specified in propositional logic. As more rules are added to the rule set it is possible that some rules become inconsistent with new rules being added. Thus, we check consistency of the rule set using boolean satisfaction and optimization library Sat4j (Anonymous, 2016) every time a new rule is added. The rule engine entails a simple and user-friendly interface connected to the web-based subsystem.

Visual analytics: We believe that our system will grow to become a rich source of information in due course as more valuable data is recorded. Thus, analyzing recorded data to discover hidden knowledge by means of visual data analytics would definitely provide a great business value to system stakeholders. Visual analytics focuses on reasoning analytically by utilizing visual inferences.

Many song artists would be intrigued to know about the popularity of their songs, the trends they currently have established and to make comparisons between themselves and the others in the industry. The ability to visually analyze information for these intents would definitely help them in making important decisions about the pieces of art they produce for competitive advantage. Similarly, radio stations may also benefit from using our visual analytics capabilities. Any song artist or radio station representative can formally request system owners to use visual analytics tools of the system. Users may be charged a small fee for the service provided. This could vary with the rules and regulations governing the system. For instance if the user belongs to a particular accredited association of song artists, the fee could be lower for him/her as opposed to a non-member. An association following this business model may well increase their member base due to these added benefits. We provide two types of visual analytics to users. They are:

- Simple dashboard analytics
- Advanced business analytics

The main purpose of simple dashboard analytics is to give the user an overall insight into the current situation of the monitoring process. They will be able to obtain an overall idea about how many songs are broadcast daily and so on. Due to space constraints, we herei n discuss only about system's ability to reason about general emotions conveyed by a song. The type of emotion of a particular song is recorded in the system at the time of song registration (done by system owners). The registrar can tag a song with particular emotion (s) from a given set of basic emotion categories. Broadcast details combined with emotion data can be used for instance to understand the mentality of the community at large at certain times of the day, week, month, year, etc.

In the advanced business analytics component several useful functionalities are provided to support businesses. Trend comparison among artists and radio stations, song popularity metrics and complete analysis of hits for a given song are among them. We note that all the analytics capabilities provided by the system are customizable. A screenshot of the system with analytics tools is shown in Fig. 4.

RESULTS AND DISCUSSION

Experimental evaluation: Due to space constraints, we only present details pertaining to experimentally evaluating our novel audio monitoring approach over an 8 h long radio broadcast stream which consists of 71 songs. This stream was generated by combining individual broadcast streams from four different FM radio stations in Sri Lanka. On the other hand our song database contained 3000 songs at the time of the experiments. Our approach was implemented using Java for computation on the main memory. We conducted experiments on an Intel (R) Core i7-7920HQ/64bit with 16 GB of RAM, running Fedora 25. The network speed was 1GB ps. Table 1-3 show details of audio content of the 8 h long stream.

Radio stations usually make changes to songs before broadcasting them. These changes include adding audio watermarks, remixing, alterations to playback speed, cover versions, etc. However, according to our independent analysis of songs broadcast through radio, 90% of the radio stations in Sri Lanka broadcast 94%

Table 1:Object type duration of 8 hour long radio broadcast stream

Object type	Duration (min)
News	40
Songs	213
Advertisements	237

Table 2: Song type counts of 8 h long radio broadcast stream

Song type	No. of songs
Cover songs	3
Songs in other languages	4
Sinhala original	64

Table 3: Song identification results

Variables	Song is in both stream and database	Song is in the stream, not in the database
Identified	46	0
Unidentified	19	6

Table 4: Diagnostic measures. TP = True Positive, TN = True Negative, FP = False Positive, FN = False Negative

TP rate	FP rate (%)	TN rate (%)	FN rate (%)	Precision (%)	Recall (%)	Accuracy (%)	F-measure(%)
69.23%	0	100	30.77	100	69.23	73.23	81.81

original songs and 6% cover versions, remixes or songs with different playback speeds. Therefore, we claim that the 8 h long radio broadcast stream we use here to best represents the population of songs aired through radio stations in Sri Lanka.

Accuracy evaluation: We evaluate the accuracy of our approach for the case where some of the broadcast songs are not already registered in the database. Table 3 shows the number of songs identified and unidentified. For our 8 h long sample stream, the approach thereby yields a 73.23% accuracy in identification of Sinhala original songs.

Diagnostic measures shown in Table 4 in relation to the accuracy evaluation, demonstrate a 0% false positive rate which is highly important for a royalty payment system where monetary rewards are exchanged between two external parties (i.e., from radio station admins to rights holders). Moreover, the manual verification of false positives would comparably be harder in our context. Furthermore having a 30.77% false negative rate is much better than having even a smaller false positive rate in the sense that false negatives can always be compensated through manual intervention. Anyhow, we have implemented a separate user Interface with a comprehensive process flow on the web-based subsystem to manually identify true negatives and false negatives that were unidentified in the automated process. It is evident that identification of true negatives require registering respective songs in the database. From another perspective, existence of a 0% false positive rate demonstrates the fact that advertisements, news and other non-song objects are trivial cases for our approach and it is effectively resistant to such kind of impurities in the stream. We note that achieving 0% false positive rate is a significant novel contribution of this study and handling of the immediate compensation of accuracy with the aid of a comprehensive process flow is of paramount importance to the correct behaviour of the system.

Since, the number of negative results can extensively supersede the number of positive results, a simple accuracy measure in percentage may not be desirable in our context. Therefore, F-measure which considers both precision and recall in its computation can be used to provide an overall quality metric about the performance of the approach. We note that our approach demonstrates 81.81% F-measure which is yields a significantly better performance than that of (Senevirathna, 2013) when (Senevirathna and Jayaratne, 2013)'s approach is executed over the same sample stream.

Table 5: Song identification results for each song type

Song types	No. of songs identified	No. of songs unidentified
Cover songs	0	3
Songs in other languages	0	4
Sinhala originals	46	18

Table 6: Song identification details with noise

Variables	Song is in both stream and database	Song is in the stream not in the database
Identified	46 (Not changed)	0 (Not changed)
Unidentified	19 (Not changed)	6 (Not changed)

Cover song identification: We have also, evaluated our approach against identification of cover songs of those originals present in the database already. A cover song is a new recording of a previously commercially released song. As shown by the results in Table 5, all cover songs were unidentified there by confirming that our approach needs major improvements for identification of them.

Noisy or distorted song identification: We added white noise uniformly to the sample stream to evaluate our approach against noisy receiver channels. White noise synthesized by utilizing 20% intensity from the maximum amplitude was added to the radio stream and the results are shown in Table 6. It is evident that no change was detected in results in presence of noise and therefore, one can conclude that our approach is robust to the additive white noise and performs equally well as in noiseless environments.

Watermarked song identification: Radio stations usually tend to add their names, sponsor details, etc., aswatermarks in to all or some song broadcasts. We evaluated the effects of audio watermarking on our identification approach. The sample stream we used consisted of eight songs with watermarks added in to them. We found that all watermarked songs in the stream were successfully identified by our approach.Furthermore, results showed a 100% success rate for identifying such songs even with added white noise.

Non-song objects amidst a song broadcast: It is customary that radio stations include talks or advertisements while airing a song for its agreed duration. Figure 8 shows two usual ways of adding non-song objects amidst a song. In case 1, the advertisement is played separately from the song possibly pausing or muting the song while in case 2 it is played with a lower intensity while the song progresses. Our results confirm a 100% success rate in both cases.

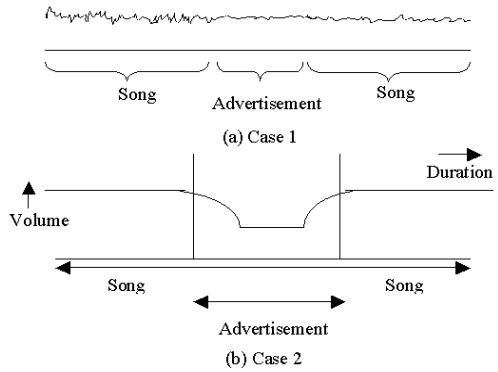

Fig. 8: +Playing non-song objects amidst a song: a) Case 1 and b) Case 2

Table 7: Processing times of various sized broadcast audio streams

Duration of the clip	Time
1 h 30 min	40 min
8 h	3 h 50 min
24 h	10 h 20 min

Efficiency evaluation: We regard that the time taken for the audio monitoring process (esp. identification process) is highly important for the overall function of the broadcast monitoring system since every 24 h radio broadcast stream from a radio station needs to be processed before the next day's stream is received. Table 7 shows the processing time taken for the identification process. Therefore, one can efficiently process a 24 h broadcast audio stream in 10 h and 20 min with our approach as opposed to that of (Senevirathna and Jayaratne, 2013) which takes more than a day to process a stream of this duration.

CONCLUSION

This study described an automated radio broadcast audio monitoring system that is capable ofcontinuously filtering and identifying songs broadcast through a known set of radio stations, storing broadcast information, generating payment reports for rights holders and visually analyzing the useful statistics related to song broadcasts. According to experimental evaluation of the audio monitoring approach it can be concluded that the audio monitoring subsystem exhibits a favorable accuracy with zero false positives which is ideal for a royalty payment processing system. The system also provides a manual solution to handle any true or false negatives and hence can be denoted as a comprehensive royalty processing system. The experimental results further demonstrated that the approach is resistant to white noise, audio watermarks and non-song objects.

RECOMMENDATIONS

We plan to extend this work to broadcast audio domains other than radio (e.g. TV, web, etc.). Our audio monitoring approach can be extended to identify cover songs on different pitches with different playback speeds, sung by different artists as well as remixes or to group songs according to their genres for better data analytics. In order to improve the process flow for unidentified songs, one may extend our approach with capabilities to identify the singer of an unidentified song. The system can further be improved to provide functionalities to predict for example, the top ten playlist for the next month. An artist can also be given the capability to predict the popularity that a particular song will receive in the next few months to come.

ACKNOWLEDGEMENT

This research was supported by a research grant awarded by Outstanding Song Creator's Association (OSCA) of Sri Lanka.

REFERENCES

Anonymous, 2016a. Automatic Content Recognition (ACR) cloud services. ACRCloud, New York, USA. https://www.acrcloud.com/

Anonymous, 2016b. Explore our products. Amazon Web Services, Inc. Seattle, Washington, USA. https://aws.amazon.com/

Anonymous, 2016c. Frequencies for equal-tempered scale,A4 = 440 Hz. Michigan Technological University,Houghton,Michigan.https://pages.mtu.edu/~suits/notefreqs.html

Anonymous, 2016d. MongoDB Atlas database as a service. MongoDB Inc., New York, USA. https://www.mongodb.com/

Carlson, A.L., 2011. Automatedmediaandcontentreporting system forbroadcastmedia.US20110015968A1, NewYork,USA.https://patents.google.com/patent/US20110015968

Haitsma, J. and T. Kalker, 2002. A highly robust audio fingerprinting system. Proceedings of the 3rd International Conference on Music Information Retrieval (ISMIR'02), October 13-17, 2002, IRCAM, Paris, France, pp: 107-115.

Le Berre, D. and A. Parrain, 2010. The sat4j library, release 2.2, system description. J. Satisfiability, Boolean Model. Comput., 7: 59-64.

Lima, I.R., T.D.C. Freire and H.A.X. Costa, 2012. Adapting and using scrum in a software research and development laboratory. Salesian J. Inf. Syst., 9: 16-23.

Miloslavsky, A., C. O'Kelly, J. Feldman and M. Koifman, 2010. Shared royalty platform for content royalty management. US7747474B2, EquaTrax, LP., New York, USA. https://patents.google.com/patent/US7747474.

Senevirathna, E.N.W. and K.L. Jayaratne, 2013. A highly robust audio monitoring system for radio broadcasting. GSTF J. Comput., 3: 1-12.

Senevirathna, E.N.W. and L. Joyaratne, 2015. Audio music monitoring: Analyzing current techniques for song recognition and identification. GSTF. J. Comput., 4: 23-34.

Wang, A., 2003. An industrial strength audio search algorithm. Proceedings of the 4th Annual International Conference on Music Information Retrieval (ISMIR'03), October 27-30, 2003, Johns Hopkins University, Baltimore, Maryland, pp: 7-13.