Hindawi Publishing Corporation
EURASIP Journal on Audio, Speech, and Music Processing
Volume 2010, Article ID 572571, 12 pages
doi:10.1155/2010/572571

Research Article

Efficient Advertisement Discovery for Audio Podcast Content Using Candidate Segmentation

M. N. Nguyen,[1] Qi Tian,[2] and Ping Xue[1]

[1] *School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798*
[2] *Institute for Infocomm Research, 1 Fusionopolis Way, Singapore 119613*

Correspondence should be addressed to M. N. Nguyen, nhutnguyen@ntu.edu.sg

Received 20 November 2009; Revised 16 April 2010; Accepted 29 June 2010

Academic Editor: Yannis Stylianou

Nowadays, audio podcasting has been widely used by many online sites such as newspapers, web portals, journals, and so forth, to deliver audio content to users through download or subscription. Within 1 to 30 minutes long of one podcast story, it is often that multiple audio advertisements (ads) are inserted into and repeated, with each of a length of 5 to 30 seconds, at different locations. Automatic detection of these attached ads is a challenging task due to the complexity of the search algorithms. Based on the knowledge of typical structures of podcast contents, this paper proposes a novel efficient advertisement discovery approach for large audio podcasting collections. The proposed approach offers a significant improvement on search speed with sufficient accuracy. The key to the acceleration comes from the advantages of candidate segmentation and sampling technique introduced to reduce both search areas and number of matching frames. The approach has been tested over a variety of podcast contents collected from MIT Technology Review, Scientific American, and Singapore Podcast websites. Experimental results show that the proposed algorithm archives detection rate of 97.5% with a significant computation saving as compared to existing state-of-the-art methods.

1. Introduction

Podcasting is already an important Internet application with millions of subscribers and is still growing rapidly towards a projected audience of 56 million by the year 2010 [1], it results in massive collections of audio podcasting contents available on the Internet. Most podcast content is audio in the form of news feeds, interview transcripts, entertainment, and radio shows. It is often that multiple advertisements (ads) are inserted into one podcast story at different locations to advertise for publishers or sponsor companies. The increasing amount of audio podcasting creates the need to develop algorithms and systems for users to organize, storage, personalize and especially search the audio podcast content collections.

Recently, audio advertisement discovery and detection systems have attracted attention of researchers due to their important role in many practical applications that are demanded by both publisher and listener. From the viewpoint of end users, who have few options to determine what ads they want to listen on the podcast channels, automatic filtering of repeated and boring ads provides better experience and higher efficiency. On the other hand, podcast advertisers and publishers are seeking cost-effective ways to replace outdated advertisement in the rebroadcasts with the new one for business purposes or target individual viewer's interests and preferences. This ad replacement increases the value to both distributor and viewer. When podcast material is redistributed by individual request, the original advertisements can be removed and replaced with new ads that are more tightly targeted to listeners. Moreover, for the marketing analyzers, and advertising planers, automatic monitoring of ads can help to collect informative statistics on the distributed advertisements. However, information about the original distributed podcast ads and their insertion points are rarely available. This creates the need to efficiently and accurately discover and segment advertising material out of the podcast contents.

In this paper, an efficient advertisement discovery approach for audio podcasting is proposed to automatically

discover and locate repeated advertisement for huge audio podcasting collections. Compared to existing work, the proposed approach has major contributions as follows. We first present typical audio podcasting structures and common temporal podcast distributions as well as podcast advertisement characteristics. Based on this knowledge, a Repeated Ad Database is built on the fly and the characteristics of ads selected from this database are used to train a fuzzy neural network classifier, which is used as a candidate segmentation to preprocess the audio collection data and select candidate segments for searching. This approach not only helps to reduce a significant portion of search area but also limits the search to shorter buffer length.

Second, we propose a multistage approach to greatly accelerate search speed using two stage detectors in cascade by combining both detection and discovery-based techniques. In the first stage, by making use of the knowledge about typical structures of podcast contents, candidate segmentation using fuzzy neural network is used to quickly narrow down search areas. In the second stage, a sampling technique is employed to discover new and unknown advertisements. Finally, we analyse the computation savings in a mathematical way and present in detail the tradeoff between search speed and detection accuracy. As compared to existing state-of-the-art methods and simple brute force, our proposed approach greatly improves search speed by 20 times to 100 times, respectively, while maintaining a high detection rate of 97.5% and obtaining the lowest false detection rate.

The rest of the paper is organized as follows. Related work is given in Sections 2, and Section 3 presents typical audio podcasting structures together with common temporal distributions of podcast contents. The proposed approach and detailed computation analyses are described in Section 4. Finally, experimental results are shown in Section 5 followed by the conclusions in Section 6.

2. Related Work

Advertisement detection and discovery for large audio collection data are crucial problems that are highly related to audio processing techniques, such as segmentation, indexing and retrieval. Existing advertisement extraction approaches can be classified into two categories. The first one, *detection-based approach,* makes use of given clues or particular features of certain classes of the advertisement, such as black/silent frames and difference on audio volume, to detect and locate advertisements [2–4]. The second one, *discovery-based approach*, makes use of repetition detection methods to discover new and unknown ads from an existing collection and keep a database of all these advertisements for matching and detection of their recurrences [5–8]. In the first approach, feature extraction and matching are performed for advertisement detection. In the second approach, it needs to discover ads from an existing collection and keep a database of all known advertisements. Depending on the needs of particular applications, either one of the two approaches can be employed. However, both techniques face the same challenges of accurate, robust detection and computationally efficient implementation for massive amounts of audio/video data to be processed.

An interesting model used to detect a given known set of advertisements is proposed in [2]. In their method, given a library of advertisements, they calculate a fingerprint for a sequence of frames based on the Color Coherence Vectors. The fingerprints are then compared to detect recurrence of known advertisements. The authors pointed out that any fingerprint that is tolerant of channel deformations has low dimension and discriminates well between different advertisements will suffice. The main advantage of this approach comes from its high efficiency of detection, which allows it could be applied to real-time advertisement recognition applications. Another example of the techniques in this category could be found in [4], in which the detection-based technique can also be used as a filtering method to accelerate the advertisement recognition [4]. In addition, techniques based on segmentation and classification, which extract acoustic features and use classifier modes such as SVM or HMM to classify each clip into advertisement or program, have been proposed to provide a general solution for advertisement classification [9].

The techniques in the first category have the advantage of efficient and fast detection due to their low-computation search algorithms; however, these techniques suffer from the following drawbacks. The clues used to detect advertisement are not always reliable for general applications. Moreover, advertisements are nowadays becoming extremely similar to normal program contents. As a result, particular features could not help to distinguish properly advertisement from other program contents. When black/silent frames are not used at the beginning and end of some advertisement breaks, this type of techniques will fail. Moreover, a general threshold that is suitable for different broadcast channels and programs is very difficult to find; therefore, detection-based approach is very sensitive to the broadcasting.

On the other hand, the techniques in the second category can overcome the above problem by using a repetition detection approach, which does not make any assumption about the location or nature of the advertisement, resulting in universality and robustness. However, the major problem of these techniques when applied to large collection data is the high computation required of the matching strategy. While the detection-based techniques can be performed by constantly matching the fingerprints of a known library of advertisements with every incoming signal, the case of discovery an unknown advertisement is more complex. In [7], the authors reported an efficient short video repeat identification based on similarity fingerprint matrix together with the locality-sensitive hashing which are usually used to reduce search complexity. This approach pays a price of computation cost as it performs an exhausted search for the whole data stream. In order to overcome the exhausted search, a sampled search algorithm was introduced in [6]. Rather than checking every single frame, the authors proposed an approach to operate only on a small sample set of the stream. However, in their approach, they used fixed write and check rates, which may affect to the detection

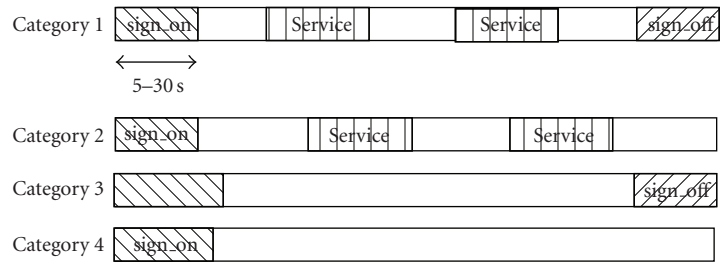

Figure 1: Structures of podcast content and advertisements.

accuracy if any failure happens in sampled frame extraction or matching. Moreover, their search is applied to the whole stream, which leads to a very much computation being required when dealing with a large collection data.

Besides the computation consuming, another issue that discovery-based techniques need to handle is how to manage the search buffer effectively when dealing with a huge collection. In [8], the authors introduced an ARGOS system for repeated object extraction and showed that it is not necessary to buffer large sections of the stream. Although the detection process of the ARGOS system, which makes use of detected library of repeated objects, significantly increases the search efficiency, its discovery process still suffers from the exhaustive search over all the collection data.

All the proposed systems in [6–8] have in common with our work that they do not seek particular features of certain classes of the advertisement; for example, the advertisements are often followed by two blank/silent frames. The contrastive point between our work and many previous approaches is that they focused on very long unstructured streaming sequences, while our work is applied to multiple short audio podcast contents and makes use of the knowledge about their typical structures to quickly narrow down search areas. Another point of difference is that while other approaches either employed audio as a preprocessing for video detection or subsequently used video features to verify the extracted objects, our approach considers solely audio features for audio podcast advertisement extraction.

3. Audio Podcast Structure

Audio podcast contents can be in various compressed audio formats such as mp3, advanced audio coding (AAC). They are usually short and less than 30 minutes long for each content item. It is often that multiple audio ads are inserted into one podcast story at different locations, ranging from 5–10 seconds to 20–30 seconds for each advertisement.

Typically, audio podcast items can be classified into 4 different categories which consist of three ads types; they are sign-on, service, and sign-off as shown in Figure 1. *Sign-on* ads, which are usually inserted at begin of the podcast items, contain information of the publishers such as name and frequency. *Service* ads, which are usually inserted somewhere inside the story, are used for sponsor company advertisements. On the other hand, the *sign-off* ads can be used for either publishers or sponsor companies, or just simply used as stop sign.

In general, *sign_on* and *sign_off* advertisements' locations are fixed at the start and the end of the podcast items, respectively. In some sense, this knowledge could help the detection process to be easier. However, multiple *service* ads could be inserted anywhere along the podcast story. Therefore, in the discovery process, we do not assume any location assumption as well as the nature of the advertisement. This location information is only used to classify and identify advertisements after they are extracted by our proposed search algorithm.

Figure 2 shows a typical temporal distribution of an audio podcasting content with a length of a few minutes. The inserted advertisement is usually begun and followed by some music scenes, and the advertisement, itself, also contains some short music to attract the listeners' attention. So we can assume that the advertisement segment could be differentiated from a "pure talk" segment based on a certain music-like or energy difference on some frequencies. These assumptions do not help to exactly identify and locate the ads, as some parts of the podcast program also contain high level of background music. However, by making use of this knowledge, we can quickly reduce a significant portion of search area by skipping all the "pure talk" segments, denoted as noncandidate segments, which have low music level and "surely" do not contain any advertisement. The detailed descriptions of the proposed approach are given in the next section.

4. Sampled- and Skipping-Search Approach

The most challenging tasks in automatically discovering a repeated advertisement are how to manage buffering efficiently and handle the complexity of the search strategy when dealing with a huge collection data. Different from audio detection techniques that seek recurrences of known advertisement, discovery of unknown repeated ads is more complex. Detection of known advertisement only requires comparing each incoming audio frame against a library of known advertisement, which is usually small as compared to the huge collection of sought data. Therefore, the complexity of the detection problem is linearly related to the size of the collection. On the other hand, in discovery problem, we must first find out what the repeated ads are, and then detect all of their recurrences. Thus the computation required

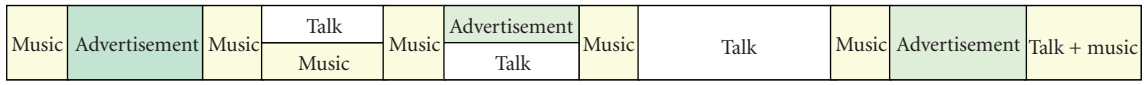

Figure 2: Illustration of audio podcast distribution.

for discovering unknown advertisement grows exponentially with the search length as well as the size of the collection data.

Assume that we search for a podcast collection of total N files resulting in the length of $L = \sum_{i=1}^{N} li$ which has L/Fs nonoverlap frames, where li is the length of ith file, and Fs is the frame size. For a simple search, every frame will need to match against $N_B = \sum_{i=1}^{n_B} li \times 1/Fs$ frames of a Buffer window containing of n_B files. Therefore, the total comparisons required would be

$$\frac{L}{Fs} \times \left(\sum_{i=1}^{n_B} li \times \frac{1}{Fs} \right) = \frac{L}{Fs} \times N_B. \tag{1}$$

From (1), there are two factors, the search length L of the collection and the number of matching frames N_B of the Buffer window, that determine the complexity of the search algorithm. The fact is that an exhausted search for every frame for the whole collection is not necessary. The efficiency of the search could be improved by reducing the number of frames to be matched, or shortening search areas.

In this work, we propose an approach, namely, Sampled and Skipping Search (SSS), to discover and identify unknown advertisements for large collection of audio podcasting by making use of specific knowledge of typical podcast structure consisting of multiple short files. The proposed approach enjoys advantages of efficient sampling technique and candidate segmentation to offer significantly fast search with sufficient accuracy by subsequently reducing both the search length and the number of matching frames.

The proposed SSS approach is shown in Figure 3 in which a Repeated Ad Database is used to store a library of found ads in order to avoid the need to buffer a long data stream. Note that this database is not given, but it is built on the fly. Making use of the audio podcast knowledge presented in Section 3, the audio podcast files of the collection are first preprocessed by the candidate segmentation in which a fuzzy neural network is trained by advertisement templates selected from the Repeated Ad Database to classify the input podcast signal into one of two classes (candidate or noncandidate). Candidate segments are the segments that have highly likelihood to be advertisement templates, while noncandidate segments have very low-likelihood values. Next, only the candidate segments are feed into the Repeated Ad Detection, which employed a sampling search technique to efficiently detect new and unknown ads. Their boundaries are refined and classified as *sign_on*, *sign_off*, or *service* ads based on their reference locations in the podcast items. These identified ads are then added into the Repeated Ad Database for further detection of their recurrences.

4.1. Candidate Segmentation. In this section, making use of audio podcast knowledge, a fuzzy neural classifier is employed to quickly narrow the search area of the huge collection data. The input signals of podcast files are labeled as candidate segment, which has advertisement characteristics such as music, high-peak signal, or silence break, or noncandidate segment, which is more concerned to "pure talk" segment. In other words, the candidate segment is an audio segment that has higher probability of advertisement appearance, while the later is an audio segment that has low probability or does not contain any advertisement.

4.1.1. Feature Extraction and Analysis. In the audio podcasting, the signal mainly comes from speech, music, and environment. Therefore, given the audio podcast files, we first uniformly segment them into nonoverlapping 1-s clips. Then, 8 features both temporal and spectral, which are chosen to represent each segment, are extracted to capture the structure of difference advertisements as candidate segments. They are energy-entropy block (EEB), short-time energy (STE), low-STE ratio (LSTER), short-time zero-crossing rate (ZCR), high-ZCR ratio (HZCRR), Spectral Roll-Off point (SRP), Spectral Centroid (SC), and Spectral Flux (SF). The detailed descriptions are given as follows.

In temporal domain, the mean and variance of STE and ZCR are common features that have been widely applied in speech analysis. The short-time energy computed for every 20 ms audio frame with 10 ms overlapping is defined as the sum of squares of the signal samples normalized by the frame length F,

$$\mathrm{STE} = \frac{1}{F} \sum_{t=1}^{F} X_{(t)}^2. \tag{2}$$

Energy-entropy block is calculated by standard deviation of the energy entropy over a 1-s clip. While STE and EEB provide a convenient representation of the signal's amplitude evolutions over time, it is found that there are more quiet frames in pure speech (nonadvertisement) segments [10]. Therefore, the ratio of "low-energy" frames (LSTER), whose STE values are less than 0.5 of the mean value to the total number of frames within a 1-s clip, can be used to detect the nonadvertisement segment.

ZCR is calculated by counting the number of times that the time domain signal crosses zero within every 20 ms audio frame as well,

$$\mathrm{ZCR} = \frac{1}{2} \sum_{t=1}^{F} \left| \operatorname{sgn} X_{(t)} - \operatorname{sgn} X_{(t-1)} \right|, \tag{3}$$

where

$$\operatorname{sgn} X_{(t)} = \begin{cases} 1, & X_{(t)} \geq 0, \\ -1, & X_{(t)} < 0. \end{cases} \tag{4}$$

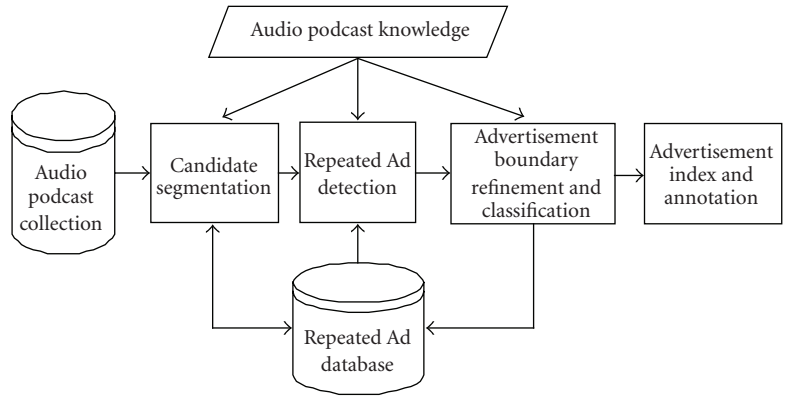

Figure 3: Overview of the proposed sampled and skipping search framework.

Although ZCR provides a reasonable way to measure the signal's frequency content of sine wave, in [11] the authors found that the variation of ZCR is more discriminative. Therefore, high zero-crossing ratio (HZCRR) is used in our experiments, and it is defined as follows:

$$\mathrm{HZCRR} = \frac{1}{N}\sum_{n=1}^{N}\left[\mathrm{sgn}\left(\mathrm{ZCR}(n) - 1.5 * \mathrm{avg}(\mathrm{ZCR})\right)\right], \quad (5)$$

where N is the number of frames in 1-s clip.

In the spectral domain, spectral roll-off point (SRP) is often used as an indicator of the skew of the frequencies presented in 1-s clip. It is defined as the frequency where 85% of the energy in the spectrum is below this point. It captures spectral shape information. This feature is used to differentiate voiced from unvoiced speech,

$$\sum_{fq=0}^{\mathrm{fro}} |X[fq]| = 0.85\sum_{fq} |X[fq]|, \quad (6)$$

where the roll-off point, fro, is the largest value of frequency fq for which the above equation is satisfied.

The spectral centroid (SC) is a measurement used in digital signal processing to characterize a spectrum. It is calculated as the weighted mean of the frequencies present in the signal, with their magnitudes as the weights,

$$\mathrm{SC} = \frac{\sum_{n=1}^{N} f(n)X(n)}{\sum_{n=1}^{N} X(n)}, \quad (7)$$

where $X(n)$ represents the magnitude of frames in 1-s clip and $f(n)$ represents the center frequency of that frame.

Spectral Flux (SF) is defined as the spectral correlation between two adjacent frames in a 1-s clip to capture the local spectral change [12]. It is effective in distinguishing certain audio types of environmental sound.

4.1.2. Candidate Selection by FCMAC-BYY Classification. In this section, the FCMAC-BYY network [13] is employed to classify the input audio podcast signal into one of two classes (candidate or noncandidate) due to its fast learning and simple computation capabilities. First proposed by Albus [14], the original CMAC has been widely applied in many areas such as robotic control, signal processing, and pattern recognition. Recently, Bayesian Ying-Yang (BYY) learning and fuzzy logic have been successfully integrated into CMAC to propose the FCMAC-BYY network, which has shown advantages in classification and regression problems [15].

As shown in Figure 4, the FCMAC-BYY network has a hierarchical structure of five layers: input layer, fuzzification layer, association layer, postassociation layer, and output layer [13].

The input to FCMAC-BYY is a nonfuzzy data vector corresponding to a measure of the input parameter represented in the respective dimension. The fuzzification layer maps input patterns into the fuzzy clusters c in the association layer through BYY learning. Thereafter, the association layer associates the fuzzy rules with the memory cell and tries to imitate a human cerebellum. The logical *AND* operation is carried out in this layer to ensure that a cell is activated only when all the inputs associated with it are fired. The association layer is then mapped to the postassociation layer where the logical *OR* operation will fire those cells whose connected inputs are activated. For the output layer, the defuzzification centre of area (COA) [16] method is used to compute the output of the structure.

In contrast to the conventional clustering algorithms, which are "one-way" learning, BYY harmonizes the training input and the solution/clusters by considering not only forward mapping from the input data into the clusters, but also the backward path from the obtained clusters to the input data. With the introduction of the BYY learning algorithm, FCMAC-BYY has higher generalization ability because the fuzzy rule sets are systematically optimized by BYY. Recently, incremental learning with sliding window has been introduced into FCMAC-BYY to dynamically construct fuzzy rule sets for time series applications [17]. In this respect, the FCMAC-BYY model becomes an ideal classifier for our candidate segmentation, which requires fast learning and adaptive and dynamic real-time training since the entire data set no longer needs to be obtained during the prediction process.

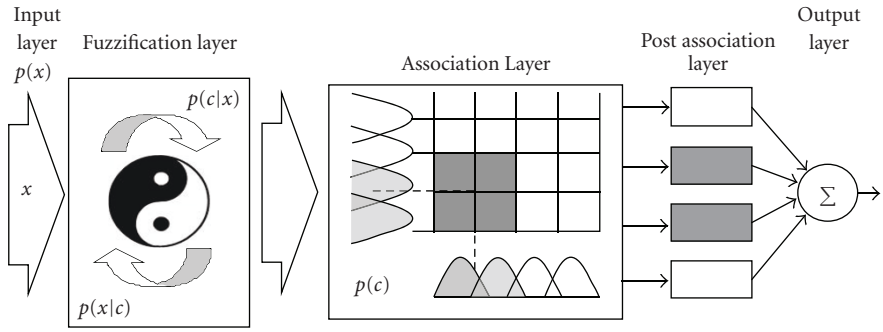

Figure 4: Block diagram of FCMAC-BYY.

In this research, the FCMAC-BYY classifier is trained by 8 features that represent for every 1-s clip from advertisement templates selected from the Repeated Ad Database. One output is used to differentiate between advertisement and nonadvertisement clips. The advertisement clips are denoted with output "1" and the nonadvertisement clips are identified by output "0". We define the candidate ratio (CR) as the ratio of the length of candidate segments to the total length of the podcast files. Then, the classification threshold of the trained FCMAC-BYY is adjusted to minimize this ratio while keeping the misclassification rate of an advertisement clip into noncandidate segment as low as zero percent.

Figure 5 shows an example of candidate segmentation results of actual podcast contents. Notice that we do not wish to exactly identify the advertisement's location in this step; however, the candidate segmentation step is only used to identify the search areas for the Repeated Ad Detection as candidate segments and skip all the noncandidate segments which do not likely contain any advertisement. In other words, the objective of this step is to quickly preprocess the huge audio podcast collection and narrow down the search areas by adjusting the threshold of the FCMAC-BYY classifier.

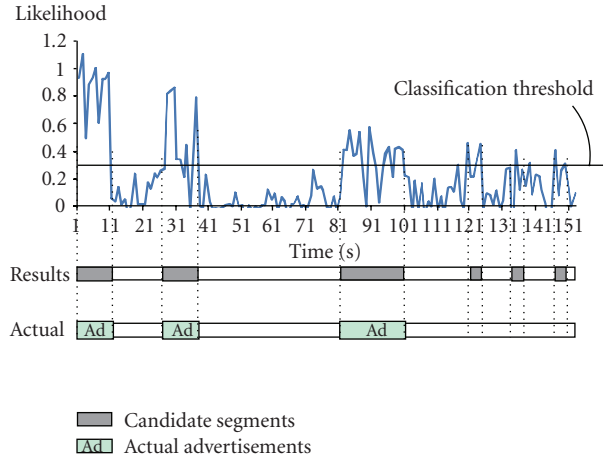

Figure 5: Example of candidate classifications by FCMAC-BYY.

4.2. Repeated Advertisement Detection. In this section, the signal of the candidate segments identified by the candidate segmentation is fingerprinted into nonoverlapping frames for discovery of unknown repeated advertisement. Once a series of candidate segments has been fingerprinted and deposited into the Buffer window, new and unknown repeated ads could be detected by two processes, *detection process* and *discovery process*, running recurrently as shown in Figure 6.

4.2.1. Audio Fingerprint Generator. In this work, the fingerprint scheme is computed based on the amount of energy in 25 audible frequency bands, known as critical bands, which lie in the range from 0 to 20 kHz for every nonoverlap interval frame size $Fs = 0.116$ seconds. The energy is calculated for each critical band and the energy difference between two consecutive bands is stored into an array of length 24. This process is done for each timeframe, resulting in a 24×1 matrix for each timeframe. Within each row of the matrix, consecutive values are compared. If the value increases from one column index to the next, the value in the prior column is replaced with a "1", otherwise it becomes a "0" as in the following equation:

$$f(r,b) = \begin{cases} 1, & \text{if } E(r,b) - E(r,b+1) \\ & \quad -(E(r-1,b) - E(r-1,b+1)) > 0, \\ 0, & \text{if } E(r,b) - E(r,b+1) \\ & \quad -(E(r-1,b) - E(r-1,b+1)) \leq 0, \end{cases} \tag{8}$$

where $E(r,b)$ is the energy of b band of frame r, and $f(r,b)$ is the bth bit of the fingerprint $f(r)$ of frame r.

4.2.2. Detection Process. This process is used to quickly detect the recurrence of the identified advertisements in the Buffer window using Repeated Ad Database. The purpose of this process is to reduce the length of the search area. From (1), one can observe that the computation increases ratio to the buffer length. By using the Repeated Ad Database to store detected advertisements, our SSS approach could limit the

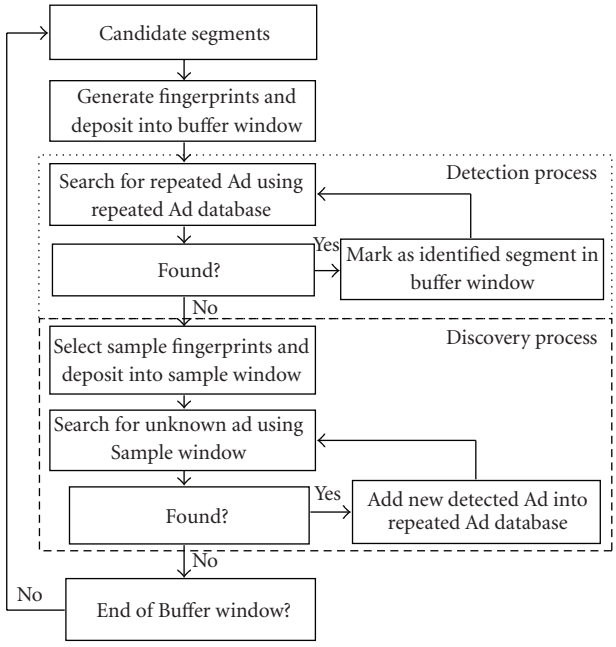

Figure 6: Flow chart of repeated Ad detection.

search to a finite distance of the length of the Buffer window. As shown in Figure 7, the Buffer window, which works as a FIFO stack to store n_B short podcast files from the collection, only has to be long enough to contain two occurrences of any new advertisement.

Once an advertisement is found by the *discovery process*, and its locations and boundaries have been identified, the clip can be added to the Repeated Ad Database as known advertisement. These detected ads are then compared with new incoming frames into the Buffer window to detect their repetitions, if a match is found, their locations are marked as *identified segment*. These identified segments do not have to be searched again by the discovery process. Therefore, the search length of the Buffer window could further be reduced by skipping all the identified segments.

The advantage of this approach is that after the second occurrence, each repeated advertisement will be added into the Repeated Ad Database and its recurrence will be quickly detected by the detection process. Every time a repetition of ads is found, we can shorten the length of the Buffer window that remains to be searched by the *discovery process*. As long as the Repeated Ad Database is smaller than the buffer, our approach improves search speed. In addition, the Repeated Ad Database of found ads can be sorted by the frequency of repetition, so that most common ads are checked first.

4.2.3. Discovery Process. This process is used to discover new and unknown repeated advertisements by comparing the fingerprints of sample window against those in the Buffer window. In this process, the sampling technique is employed to reduce the number of frames to be matched. If we assume the fact that for any frame in the repeated advertisement would have one matched frame in its repetition, we have

$$f(r1+k) \approx f(r2+k), \quad \text{for any } 0 \le k \le \text{LR}, \tag{9}$$

where LR is the length of the advertisement and $f(r1)$, and $f(r2)$ are the fingerprints at frame *r1*, *r2* of advertisement and its repetition, respectively. Note that, it is not necessary that the relative offsets, *k,* have to be exactly equal due to noise or decoding effect. As long as their relative offsets are less than a small threshold of each other, they still match.

Therefore, instead of checking the fingerprint at every single frame, the discovery process of the proposed algorithm operates only on sample frames of the buffer. As shown in Figure 7, we take only sampled frames from the Buffer window into the Sample window with the rate of one sampled frame for every *S* frames. It means that the distance between two alternate samples is $S \times Fs$. Therefore, in order to guarantee that every repeated advertisement will contribute at least one sampled frame into the Sample window, the sampling rate *S* must satisfy the following constraint:

$$S \le \frac{l_{\min}}{Fs}, \tag{10}$$

where $l_{\min}$ is the minimum length of the advertisements.

Once a match is found, we would expect that a repeat has occurred. However, the boundaries of the two repetitions have not been identified. Since the match corresponds to a repetition of the advertisement at the location of the sampled frame found, we are able to identify the starting and ending positions of the repeat by tracing the fingerprint of frames backwards and forwards, respectively.

4.3. Analysis Computation Saving. We are now in the position to analyze the improvement of the proposed SSS approach over the simple brute force approach. Assume that we search for a podcast collection of a total N files, resulting in the length of $L = \sum_{i=1}^{N} li$ which has $N_F = L/Fs$ nonoverlapping frames, where li is the length of ith file, and Fs is the frame size. For a simple search, every frame will need to match against the frames of the Buffer window containing of n_B files. Therefore, the total comparisons required would be

$$N_F \times \left(\sum_{i=1}^{n_B} li \times \frac{1}{Fs} \right). \tag{11}$$

On one hand, instead of a search for the whole podcast collection, the candidate segmentation of the proposed SSS approach narrows down the search areas by a candidate ratio CR (the ratio of the length of candidate segments to the total length of the podcast files). Therefore, the number of frames to be searched of the collection and the Buffer window size are now shown in (12) and (13), respectively,

$$\frac{L \times \text{CR}}{Fs}, \tag{12}$$

$$\sum_{i=1}^{n_B} li \times \text{CR}. \tag{13}$$

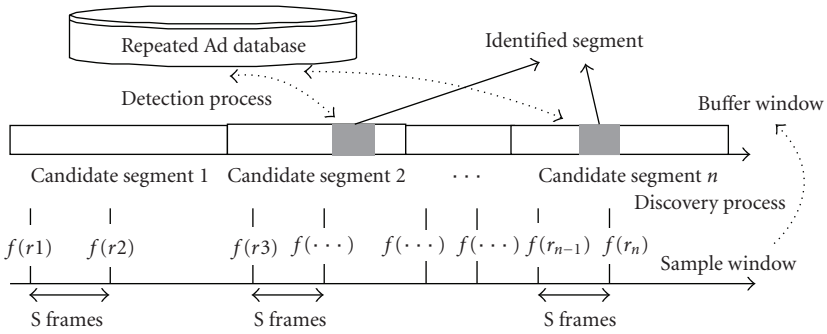

Figure 7: Detection and discovery process of the proposed approach.

On the other hand, by reducing the number of frames to be matched in the discovery process using sampling technique, the SSS approach needs to match only the sampled frames in the Sample window with the sampling rate S. Therefore, the number of matching frames required by SSS approach is

$$\sum_{i=1}^{n_B} li \times \mathrm{CR} \times \frac{1}{Fs} \times \frac{1}{S}. \tag{14}$$

From (12) and (14), we have that the upper bound of total comparisons required by SSS is

$$\left(\mathrm{CR} \times \frac{L}{Fs}\right) \times \left(\sum_{i=1}^{n_B} li \times \frac{1}{Fs} \times \mathrm{CR} \times \frac{1}{S}\right). \tag{15}$$

As compared to (1), the proposed SSS approach has saved a significant computation of S/CR^2 times on average. Therefore, the two factors, sampling rate S and candidate ratio CR, will determine the complexity of the proposed SSS approach. The detailed analyses of the effect of these two factors as well as the tradeoff between search speed and detection accuracy will be given in Section 5.

5. Experimental Results

We are now in the position of illustrating the performance of the proposed approach. The hardware configuration for our experiments is Intel Pentium IV core 2 Duo 2.66 GHz CPU using Microsoft Windows XP with 2.0 GB memory. The podcast items are collected from various websites including MIT Technology Review (MIT) [18], Scientific American (SA) [19], and Singapore Podcast (SP) [20] as shown in Table 1. The podcast collection and ground truth information could be downloaded in [21]. There are 325 audio podcast files which contain 906 repetitions (including 325 Sign_on, 264 Service, and 317 Sign_off advertisements). The collection is 1348 minutes in length.

We first show the effect of the proposed candidate segmentation on shortening the search length. This experiment is conducted in two steps as follows. In the initialization step, the Repeated Ad Database is built from the first 300 minutes of the collection using the proposed approach without candidate segmentation. The learning (training) data set is

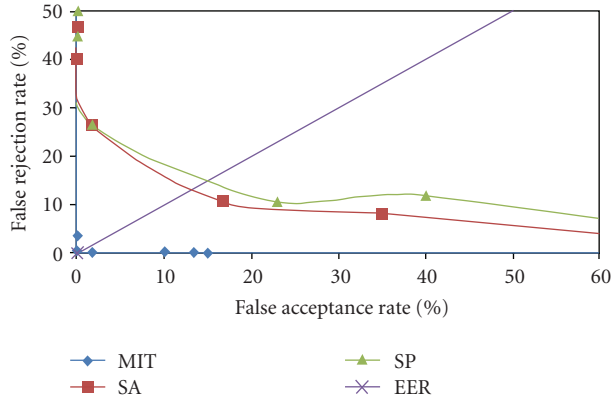

Figure 8: The ROC curves of candidate segmentation.

then selected from these detected advertisements to train our FCMAC-BYY classifier. The rest of the collection data is used in the testing step. During this step, new advertisements discovered by the *discovery process* will be added into the Repeated Ad Database and be used to update the learning data set subsequently.

The classification threshold (to discern between candidate and noncandidate segments) based on 8 features of both temporal and spectral of the FCMAC-BYY classifier [13] is varied to obtain the receiver operating characteristic (ROC) curves depicted in Figure 8. False Acceptance Rate (FAR) is defined as the error of classifying an advertisement clip as a noncandidate segment whereas False Rejection Rate (FRR) is the ratio of classification of a nonadvertisement clip as a candidate segment. The EER line denotes the case of equal error rates, where False Acceptance Rate equals False Rejection Rate.

From Figure 8, it is observed that while the advertisement clips of the MIT Technology Review podcasting could be separated easily with low error values, the advertisement and nonadvertisement clips of the Scientific American and Singapore Podcast podcasting have a significant degree of overlapping. The above results could be explained by the nature characteristics of these three podcasting websites as shown in Figure 8.

The advertisement clips of the MIT podcasting have high-likelihood values and the program (nonadvertisement)

Table 1: Summary of audio podcast collection.

	Number of files	Total length in minutes	Number of advertisements		
			Sign_on	Service	Sign_off
MIT technology review	126	570	126	126	118
Scientific American	58	333	58	13	58
Singapore podcast	141	445	141	125	141
Total	325	1348	325	264	317

clips have quite low-likelihood values, therefore, the candidate segments selected by our classifier mainly contain actual advertisement clips with low False Rejection Rate (Figure 9(a)). While, the program clips of the Scientific American and Singapore Podcast podcasting have high-likelihood values, resulting in high False Rejection Rate (Figures 9(b) and 9(c)). More detailed analyses on the candidate ratio (CR) of the proposed candidate segmentation using FCMAC-BYY are given as follows.

Table 2 shows the advertisement classification rate and the candidate segmentation percentage of the three podcast websites. The experimental results of the FCMAC-BYY classifier are compared against the SVM, which is implemented using LIBSVM package [22]. As mentioned above, due to the higher overlapping degree of advertisement and nonadvertisement classes, the Scientific American and Singapore Podcast podcasting have higher False Rejection Rate obtained by both SVM and FCMAC-BYY as compared to MIT podcasting. It results in a larger candidate segment ratio for these two websites. As compared to SVM, the proposed candidate segmentation using FCMAC-BYY obtains similar results; however, it offers much faster computation time. Moreover, unlike SVM, which is considered as a black box classifier, the proposed FCMAC-BYY classifier offers a logic reasoning framework that can extract high level semantic rules for a better human understanding. This work will be included in our future work. On average, the proposed candidate segmentation using FCMAC-BYY identifies 38% of the total length of the podcast collection as candidate segments, which are used to search for advertisement by the Repeated Advertisement Detection, and saves over 60% total length as noncandidate segments that do not need to be searched.

In the second experiment, we instrumented our experiments to study the relationship between the detection accuracy and the searching time with different sampling rate S. The proposed approach is compared to the Sampled Search [6] which applies the search to the whole collection. In this experiment, the buffer size of the proposed SSS approach is set to the length of 30 files which approximates about 2 hours long.

From Figure 10(a), it is observed that the proposed SSS approach achieves the highest detection rate of 98% with the sampling rate $S \leq 40$, while [6] is only able to reach 96% detection rate at $S \leq 20$. Moreover, when the sampling rate increases, the detection rate of [6] drops very fast, while the proposed SSS approach is still able to maintain the detection rate of 95% at $S = 100$, and 89% at $S = 150$. These results

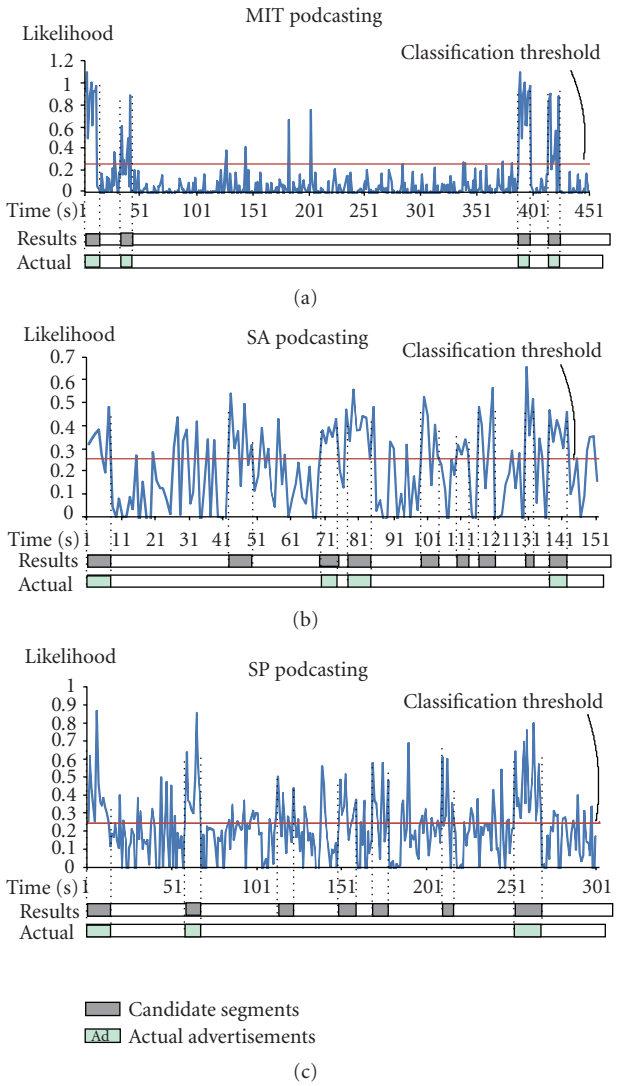

Figure 9: The sample of candidate classifications of three podcast sites: (a) MIT, (b) SA, and (c) SP.

can be explained as follows. While in [6], repeated sequences could only be identified solely based on the discovery process, SSS employs an Repeated Ad Database to store detected ads so that once a new advertisement is found, its recurrences could be identified easily by the detection process. On the other hand, by reducing the search areas, the candidate

Table 2: Candidate segmentation percentage.

		FAR (%)	FRR (%)	(CR) (%)	Total time (seconds)
SVM	MIT	0	3.26	9.33	**945**
	SA	9.33	35.53	46.24	
	SP	12.86	41.76	53.65	
	Average	**6.41**	**23.94**	**33.07**	
FCMAC-BYY	MIT	0	3.74	11.66	**173**
	SA	6.66	40.70	54.09	
	SP	8.30	47.31	59.77	
	Average	**4.38**	**27.25**	**38.02**	

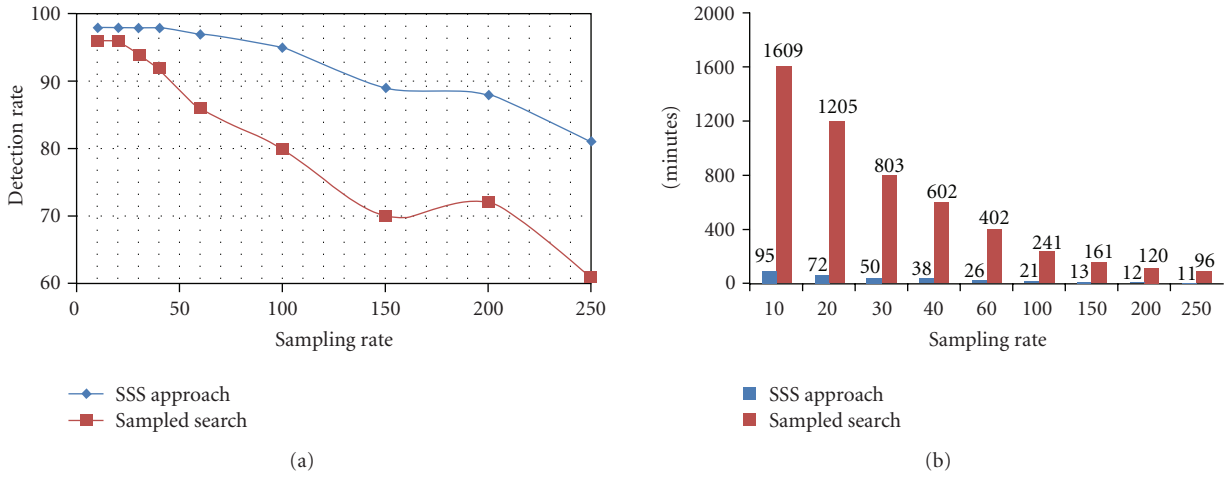

Figure 10: Illustration of the effect of the sampling rate to detection accuracy and running time.

Table 3: Illustration the effect of the size of the Buffer window to detection accuracy and running time.

Buffer size	Detection rate (%)		Search time (minutes)	
	[6]	SSS	[6]	SSS
150	92	98	602	167
100	90	98	402	113
80	86	98	321	92
60	84	98	241	70
50	82	98	201	60
40	76	98	161	49
30	73	98	120	38
20	65	95	81	28
15	61	92	40	17
10	52	86	13	10

segmentation of the SSS offers the proposed approach a faster search speed as compared to [6] as shown in Figure 10(b).

Another factor that affects to the search performance is the size of the Buffer window. Further experiments are conducted to show the effect of this factor. Table 3 shows the comparisons on detection rate and search time of [6] and the proposed SSS approach for different size of the Buffer window. In this experiment, the sampling rate is set to $S = 40$. One can observe that with the archive mechanism, SSS could achieve 98% detection rate with the buffer size of $n_b \geq 30$ files. The conclusion can be drawn is that when the size of the Buffer window reaches a certain length that is long enough to contain two occurrences of new advertisements, the proposed SSS approach is able to archive the highest detection rate while further increasing buffer size is not much helpful. This consolidates our theory presented in Section 4.2.2.

Further experiments are illustrated in Table 4 where the comparisons of detection performance on different advertisement categories described in Section 3 are given. As analyzed in the above two experiments, in this experiment the sampling rate of the proposed SSS approach and [6] is set to $S = 40$, and the size of the Buffer window of both SSS approach is chosen as $n_B = 30$ files. We measure our results using the detection rate (the ratio of number of correctly identified ads to the total number of ads) and the false detection (the ratio of number of incorrectly identified ads to the number of detected ads).

We can see that the false detection rate only falls on *service* ads; this is due to the fact that all repeated sequences located between the begin and the end of the files will be considered as *service* advertisement. It is noticed that the location information is only used to classify advertisements after they are identified by our SSS approach. On average,

Table 4: Comparison results of the proposed SSS approach.

Algorithm		Simple search	[6]	SSS
Sign_on	Detection rate	99.1%	99.1%	99.1%
	False detection	0%	0%	0%
Service	Detection rate	97.2%	96.2%	96.7%
	False detection	2.8%	2.0%	2.0%
Sign_off	Detection rate	98.4%	97.5%	98.1%
	False detection	0%	0%	0%
Average detection rate		98.3%	95.4%	**97.5**
Computation time (mins)		3696	602	**38**

the proposed SSS approach greatly improves search speed by 100 times as compared to a simple search method and 20 times as compared to [6] while maintaining a high detection rate of 97.5% and obtaining the lowest false detection rate.

6. Conclusions

This paper introduces a novel Sampled and Skipping Search approach to discover and detect unknown podcast advertisements for large podcasting collections efficiently. Based on the knowledge of typical structures of podcast contents, the proposed approach employed candidate segmentation and sampling techniques to accelerate the search speed by reducing both search areas and the number of matching frames. As compared to existing state-of-the-art techniques, the proposed approach greatly improves search speed and saves a significant computation while maintaining sufficient detection accuracy. In this paper, we show the effect of the sampling rate and the buffer size to the trade-off between detection accuracy and search speed. We also present the typical audio podcast structures and point out that buffering a large amount of files of the collection is not necessary for improving the detection rate. Finally, detailed experimental analyses conducted on a variety of podcast contents collected from various websites are reported.

Acknowledgment

This work is supported by the Agency for Science, Technology, and Research (A*Star), Singapore, under the Grant no. 062 130 0058.

References

[1] http://podcast.com/.

[2] R. Lienhart, C. Kuhmuench, and W. Effelsberg, "On the detection and recognition of television commercials," in *Proceedings of the IEEE International Conference on Multimedia Computing and Systems (ICMCS '97)*, pp. 509–516, June 1997.

[3] D. Ruinskiy and Y. Lavner, "An effective algorithm for automatic detection and exact demarcation of breath sounds in speech and song signals," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 15, no. 3, pp. 838–850, 2007.

[4] K. Kashino, T. Kurozumi, and H. Murase, "A quick search method for audio and video signals based on histogram pruning," *IEEE Transactions on Multimedia*, vol. 5, no. 3, pp. 348–357, 2003.

[5] J. Foote and M Cooper, "Audio retrieval by rhythmic similarity," in *Proceedings of the International Conference on Music Information Retrieval*, Paris, France, 2002.

[6] C. Herley, "Accurate repeat finding and object skipping using fingerprints," in *Proceedings of the 13th annual ACM international Conference on Multimedia*, pp. 656–665, ACM, Singapore, 2005.

[7] X.-F. Yang, Q. Tian, and P. Xue, "Efficient short video repeat identification with application to news video structure analysis," *IEEE Transactions on Multimedia*, vol. 9, no. 3, pp. 600–609, 2007.

[8] C. Herley, "ARGOS: automatically extracting repeating objects from multimedia streams," *IEEE Transactions on Multimedia*, vol. 8, no. 1, pp. 115–129, 2006.

[9] D. Ling-Yu, W. Jinqiao, Z. Yantao, S. J. Jesse, L. Hanqing, and X. Changsheng, "Segmentation, categorization, and identification of commercial clips from TV streams using multimodal analysis," in *Proceedings of the 14th Annual ACM International Conference on Multimedia (MM '06)*, pp. 201–210, ACM, Santa Barbara, Calif, USA, 2006.

[10] Y. Li and C. Dorai, "Instructional video content analysis using audio information," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 14, no. 6, pp. 2264–2274, 2006.

[11] L. Lie, Z. Hong-Jiang, and J. Hao, "Content analysis for audio classification and segmentation," *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 7, pp. 504–516, 2002.

[12] S. McAdams, "Perspectives on the contribution of timbre to musical structure," *Computer Music Journal*, vol. 23, no. 3, pp. 85–102, 1999.

[13] M. N. Nguyen, D. Shi, and C. Quek, "FCMAC-BYY: fuzzy CMAC using Bayesian Ying-Yang learning," *IEEE Transactions on Systems, Man, and Cybernetics B*, vol. 36, no. 5, pp. 1180–1190, 2006.

[14] J. S. Albus, "A new approach to manipulator control: the cerebellar model articulation controller (CMAC)," *Transactions of the ASME, Dynamic Systems, Measurement and Control*, vol. 97, no. 3, pp. 220–227, 1975.

[15] M. N. Nguyen, D. Shi, and C. Quek, "A nature inspired Ying-Yang approach for intelligent decision support in bank solvency analysis," *Expert Systems with Applications*, vol. 34, no. 4, pp. 2576–2587, 2008.

[16] E. S. Lee and Q. Zhu, *Fuzzy and Evidence Reasoning*, Physica, 1995.

[17] D. Shi, M. N. Nguyen, S. Zhou, and G. Yin, "Fuzzy CMAC with incremental bayesian Ying-Yang learning and dynamic rule construction," *IEEE Transactions on Systems, Man, and Cybernetics B*, vol. 40, pp. 548–552, 2009.

[18] http://www.technologyreview.com/.

[19] http://www.scientificamerican.com/podcast/.

[20] http://www.podcast.sg/.

[21] http://sites.google.com/site/nhutnguyensite/home/audio-podcast-dataset.

[22] C. C. Chang and C. J. Lin, "LIBSVM : a library for support vector machines," 2001, http://www.csie.ntu.edu.tw/~cjlin/libsvm/.