International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys.13, No.1 (May-23)

http://dx.doi.org/10.12785/ijcds/1301111

Spoof Detection using Sequentially Integrated Image and Audio Features

Nidhi Chakravarty[1] **and Mohit Dua**[2]

[1]*Department of Computer Engineering, National Institute of Technology, Kurukshetra, India*
[2]*Department of Computer Engineering, National Institute of Technology, Kurukshetra, India*

Received 12 Sep. 2022, Revised 30 Apr. 2023, Accepted 14 May. 2023, Published 30 May. 2023

Abstract: Analyzing the intricate nature of an audio signal often requires the extraction of relevant features, which serve as informative descriptors of the signal. It entails studying the signal and determining how signals are related to one another. As a result, the performance of audio spoofing detection in Automatic Speaker Verification (ASV) systems is strongly reliant on front-end feature extraction. In this paper, three types of successively integrated features have been proposed. First, Acoustic Ternary Pattern (ATP) image features are sequentially fused with different audio features such as MFCC, CQCC, GTCC, BFCC and PLP, individually. Second, LBP image features are combined with all these audio features similarly. Then, the sequential integration of ATP-LBP features is combined individually with MFCC, CQCC, GTCC, BFCC and PLP features. Finally, these front-end hybrid feature sets are classified using different ML and deep learning algorithms based acoustic models at the back-end. The state-of-the-art ASVspoof 2019 dataset has been used to implement various front-end and back-end combinations. The research outcomes reveal that the proposed approach achieved the best results with ATP-LBP-GTCC at the front end with LSTM-based acoustic model at the back-end.

Keywords: ASV, Feature extraction, MFCC, LPCC, GTCC, BFCC, LSTM, Spoof Detection.

1. INTRODUCTION

Various physiological properties of humans, such as the retina, fingerprints, and voice, can be utilized to identify a person individually. Compared to other features, voice is the most common and easy way to recognise a person. Due to recent technological improvements, speech authentication systems, such as ASV systems, have become common and prominent alternatives to conventional security systems. These systems, unlike others, do not cause discomfort or pose any health hazards to the user because there is no direct touch with the machine. According to studies, ninety percent of respondents are enthusiastic about employing audio signal-based biometrics instead of traditional ones [1]. An ASV system evaluates the speech injected through a microphone or any other recording device and approves or rejects the stated identity. The goal of speaker verification is whether a claimant's applied speech is authentic or fake. The front-end and back-end are the two main components of such systems for attaining the required functionality. As shown in Figure 1, The front end of the ASV system processes the input speech signal. In contrast, the back-end half of the system undertakes a validity check and speaker verification (by comparing stated identity with the labelled speech database) to approve or reject the stated identification.

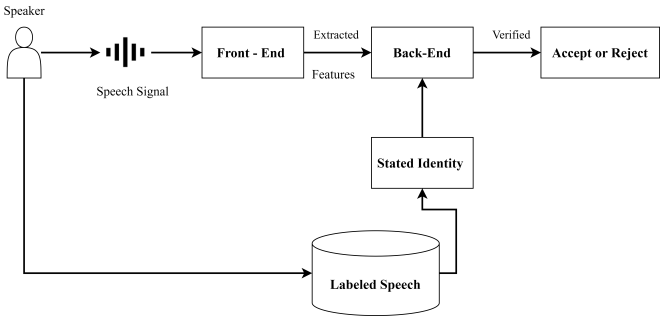


Figure 1. Components of Automatic Speech Verification System

The front-end of the system retrieves information about the speaker's uniqueness and signals authenticity, which is present in the input voice signal in the form of its characteristics [2], [3]. Feature extraction is performed in the front-end utilizing traditional approaches such as MFCC and PLP [4]. The Mel scale is utilized in MFCC, and the filter used is triangular. However, the conventional features suffer from two major limitations: additive noise and channel mismatch vulnerability. CQCC is the other popular method that is being used nowadays for feature extraction [5]. CQCC uses uniform resampling to convert the octave power spectrum into the Linear Power Spectrum (LPS). After converting into

nidhi_62000061@nitkkr.ac.in, er.mohitdua@gmail.com

LPS, Discrete Cosine Transform(DCT) is applied on LPS to obtain CQCC [6]. Also, CQCC features don't perform well in noisy environment [7]. Hence, researchers tried to modify these techniques to make these noise robust. The other approach to handle the noise during feature extraction is to use features that are already noise robust such as GTCC [8], [9] and BFCC [10], [11]. GTCC employs a non-linear gammatone filter bank[12]. Its non-linear behaviour is a key aspect in the filter's robustness in noisy environments, as it allows it to produce superior results over a wide dynamic range. BFCC uses gammachirp filter bank, which is derived from gammatone filter bank for high frequency selectivity. Another approach used by researchers to handle noise is use of hybrid feature extraction methods. A hybrid feature technique has been proposed to extract information from an audio signal by integrating different audio feature extraction techniques [11]. Recently, to make ASV system noise robust audio feature have been combined with the image features [13], [14]. The proposed work in this paper also sequentially integrates image features LBP and ATP with various audio features and tries to find the best possible combination for extraction features. The back-ends' classification model separates the processable artefacts from the applied speech features. HMM has been used as the main approach for classification at the back-end of a statistically built ASR system for many years [15], [14]. A finite state Markov chain is defined as an HMM. A probability distribution relates to each state to compute the likelihood of auditory features. However, HMM-based acoustic models are acknowledged to have various limitations such as its state contains less observation due to small training dataset which affects the robustness of ASV system and HMM has high computational cost. To overcome these problem researchers in [16], [17] suggests solutions. As speaker verification falls within the category of classification problems, ML (ML) techniques are known to be more suitable for drawing conclusions from the observed data [18]. GMM[19] and SVM Classifiers have also been used to study various modelling strategies. Acoustic observations are represented as a series of GMM vectors with discriminative SVM classification [20]. These models have been regularly employed for various ASV-related operations and are suited for processing speech-related data. However, these are ineffective for nonlinear or almost nonlinear data spread. In[21], authors used the ECOC [22] in their proposed acoustic model to create a multi-class classifier by merging three binary classifiers to distinguish genuine, first-order, and second-order replay samples in their suggested approach. The work used different ML algorithms such as KNN, NB, SVM and Ensemble Bagged Trees and bi-layered Neural network used for classification purpose. Also, due to advancements ML algorithms, in recent years, the research community has switched to deep learning models that can analyse a large dataset with complicated interactions [23].The work in this paper also exploits various ML methods based acoustic models, and LSTM based acoustic model to implement the proposed system. The rest of the paper is organized as follows. The next sub-section, 1.1 presents the related work. Section 2 briefly explains the preliminaries of the techniques used in building the proposed system, and Section 3 illustrates the proposed architecture. Section 4 gives the details of the experimental setup and results. The discussion and comparative analysis is given in Section 5, and the conclusion is described in Section 6 of the paper.

A. *Related Work*

With the advancement of technology, various types of attacks have been introduced by the adversary, which can violate human privacy [24].As discussed above, traditional features produce good outcomes in clean atmosphere, however, their performance degrades in the noisy acoustics environment. Hence, authors try to improve the performance of front end of ASV systems in noisy environment either by modifying MFCC or by using noise robust features or by using hybrid feature extraction techniques. In [25] , authors modified MFCC to improve the noise sensitivity. The logarithmic transformation in the traditional MFCC is replaced by a combination function of power, and log function. In their proposed work, Spectral Subtraction and Median-Filter also combined with the combination function to minimize the noise sensitivity. In [26], MFCC, Inverted Mel frequency cepstral coefficient (IMFCC), and Linear Prediction Cepstral Coefficients(LPCC) have been used to represent audio features. GMM with a diagonal covariance matrix is used to design a classification model. However, it did not produce a good result, so to improve the results, Deep learning models have been used such as LSTM, Convolutional Neural Networks (CNN). Chettri et al. [27] employed MFCC, IMFCC, CQCC, and Sub-band Centroid Magnitude Coefficients (SCMC) to represent audio data, and then these features have been used to train deep learning models including CNN, CRNN, 1D-Convolutional Neural Network, and Wave-U-Net, as well as ML techniques like SVM and Ensemble models also used for performance analysis. Todisco et al. [28] used CQCC feature extraction technique with two GMM, a binary classifier which was used to classify audios as genuine or spoof. Mittal et al. used CQCC with CNN, LSTM, and a combination of the static-dynamic features of CQCC with LSTM-CNN ensemble in their proposed works of [3], [29]and [30], respectively. However, the issue of noise remains open with MFCC and CQCC features. In [31], authors used GTCC feature and pitch at front-end for feature extraction, and passed these features to GMM and KNN to improve the performance of ASV system. Kaun et al. [10] applied auditory based BFCC features with AURORA 2 dataset, and compared these features' performance with MFCC using HMM model.

Noroozi et al., [32] suggested a methodology for emotion recognition using audio. They considered different features such as MFCC, pitch, variance, intensity, and filter-bank energies. In total, 88 features have been used to train the K-means and 3D CNN based classifiers at the back-end. In [33], [34] , authors suggested Local discriminant bases(LDB) and MFCC combination for information ex traction from audios. The extracted features have been fed

into a three-level hierarchical categorization of audio signals using a Linear Discriminant Analysis (LDA) based classifier. In this paper, first, the performance evaluation of both features is done individually, and then combining both the feature extraction techniques. Malik et.al. [21], represented replay attacks as a nonlinear process and proposed the ATP-GTCC[35] combination to detect the harmonic distortions. The suggested ATP-GTCC feature space is used to train a multi-class SVM classifier, and tested for audio replay attack detection using the ECOC model.

Motivated by the works of [20], [13] and [3], [29], [30], the proposed work in this paper suggests a novel integrated approach for front end feature extraction. The proposed approach uses combination of audio and image feature extraction techniques. Different types of features (image and audio) have been integrated sequentially, and a total of 15 distinct sets of integrated features have been created in the proposed work. The combinations have been created using ATP, LBP image features and MFCC, PLP, CQCC, GTCC, and BFCC audio features. The novel contributions of the proposed work can be outlined as:

- Firstly, the image feature ATP has been combined with audio FE techniques such as MFCC, PLP, CQCC, GTCC, and BFCC, individually.
- Similarly, the image feature LBP has been integrated with audio FE techniques such as MFCC, PLP, CQCC, GTCC, and BFCC individually.
- Above two steps created 10 distinct feature set. Then, these feature sets are fed to four different ML-based acoustic models such as NB, SVM, DT, K-NN and LSTM based acoustic for performance evaluation.
- After observing the result of above said dual feature combinations, we decided to create the feature set by combining both image feature techniques with all the audio feature extraction techniques. This step made 5 trio features set.
- The proposed trio features sets have been input to LSTM-based acoustic and four ML based acoustic models for classification.
- Extensive performance analysis has been done by comparing proposed models' performances using the evaluation parameters such as Precision, Accuracy, Recall, F1-score, and Equal Error Rate (EER).

2. Preliminaries

The current section discusses the front end and back-end strategies that have been used to implement the proposed work. The proposed approach uses various strategies for front end feature extraction and various different ML based algorithms for building acoustic models. The first part of this section discusses basics of visual features, the second part covers audio elements, and the third part describes the acoustic models used.

A. *Image Features*

Two types of image features LBP and ATP have been used for audio feature extraction at the front end of the proposed ASV system.

1) *Local binary pattern (LBP)*

Local binary pattern (LBP): LBP was first time introduced by Ojala et al. [36]. In these features, a local pattern is created by encoding the grey level difference between the centre pixel and its neighbours. LBP generates a M bit binary number that concatenates with the decimal numbers[37]. Unlike an image signal, where the neighbourhood is a circle covering an angle of 360o, the audio signal has only two neighbourhood angles: 0^o and 180^o . LBP allows to define an audio texture as a joint distribution of the pixel. Given a center pixel p_c with grey level g_pc , the LBP of the pixel is computed as follows:

$$y_1 = S_n[y[j+s-N/2]-y[j]]2^i \tag{1}$$

$$y_2 = S[y[j+s+1]-y[j]] \tag{2}$$

$$LBP_N(y[j]) = \sum_{n=1}^{N/2-1}\{[[y_1+y_2]-y[j]]2^{i+n/2}\} \tag{3}$$

2) *Acoustic Ternary Pattern (ATP)*

ATP technique is used to represent 1-D audio signals as the acoustic signal to detect replay attacks[38], [39]. This technique was inspired by the application of 2D-local ternary patterns in image processing[40].

The difference between the magnitudes of central sample c and nearby audio samples z^j is used to calculate the local ATP response. A threshold t_d value is used to get the most optimized features [37].

To calculate the features, firstly, signal S(x) with N_s samples is divided into non-overlapping frame F^n of length l. Secondly, sample values in F^n is quantized around the central sample C, while values above and below $C \pm t_h$ are quantized to 1 and -1, respectively. As a result, we have a three-valued function:

$$P(y_i, C, t_d) = \left\{ \begin{array}{ll} -1, & y^j-(C-t_d) \le 0 \\ 0, & (C+t_h) < y^j < (C-t_d) \\ 1, & y^j-(C-t_d) \ge 0 \end{array} \right\} \tag{4}$$

where (y^j, C, t_d) represents acoustic signal locally using a 3-valued ternary pattern. Now, all the patterns quantized to -1 are retained in P_{low}, patterns quantized to +1 are retained in P_{up}, and other patterns are replaced as zero.

$$P_{low}(y^j, C, t_d) = \left\{ \begin{array}{ll} 1 & if P\left(z^j, c, t_d\right) = -1 \\ 0 & otherwise \end{array} \right\} \tag{5}$$

$$P_{up}(y^j, C, t_d) = \left\{ \begin{array}{ll} 1 & if P\left(z^j, c, t_d\right) = +1 \\ 0 & otherwise \end{array} \right\} \tag{6}$$

Equations (7) and (8) are used to compute the upper class ATP^{up} and lower class ATP^{lw} patterns. These patterns are calculated into decimal form using following equations.

$$ATP_u^{up}(y^j,\ C, t_d) = \left\{ \sum_{j=0}^{j=7} P_{up}\left(y^j, C,\ t_d\right) * 2^j \right\} \quad (7)$$

$$ATP_u^{lw}(y^j, C, t_d) = \left\{ \sum_{j=0}^{j=7} P_{low}\left(y^j, C, t_d\right) * 2^j \right\} \quad (8)$$

Then, In the next step, histogram is calculated for ATP_u^{up} and ATP_u^{lw}. One bin of histograms is assigned for each uniform pattern and a non-uniform pattern in the bin. Functions used to calculate histogram are given as:

$$H_s{}^{up}(ATP^{up}, b) = \left\{ \sum_{k=1}^{K} \rho\ \left(ATP_k^{up}, b\right) \right\} \quad (9)$$

$$H_s{}^{lw}\left(ATP^{lw}, b\right) = \left\{ \sum_{k=1}^{K} \rho\left(ATP_k^{lw}, b\right) \right\} \quad (10)$$

where b represents the bin of histogram and ρ represents Kronecker delta function. Finally, ATP feature vector are generated by concatenating both $H_s{}^{up}$ and $H_s{}^{lw}$ using equation (11).

$$ATP = \left\{ [H_s{}^{up} \| H_s{}^{lw}] \right\} \quad (11)$$

B. *Audio Features*

Audio features are the audio signal descriptions that can be used as input into statistical or ML models. The proposed work uses Conventional features such as MFCC and PLP, and noise robust features GTCC and BFCC.

1) *MFCC*

The ideas of speech generation and perception are used by MFCC to extract acoustic information from a spoken stream. In MFCC, first, pre-emphasis operation is performed on the voice signal to increase the energy at high frequencies, followed by Hamming window function to eliminate inconsistencies and information loss. After windowing, to get high frequency feature, Discrete Fourier Transform (DFT) using equation (12) is performed on the sample.

$$f_{r,i,0} = \left\{ \left\{ \frac{1}{n} \sum_{l=1}^{N-1} \left(e^{-J2\pi \frac{l_i}{n}}\right) f_l \right\} \right\} \quad (12)$$

where, i varies from 0 to $(n/2) - 1$, and n denotes a sample point within a time frame f. Then, filtering is carried out using different kind of filters on the output spectrum obtained from DFT to measure the power spectrum as:

$$f_{r,l,1} = \left\{ \sum_{i=0}^{\frac{N}{2}-1} c_{l,i} f_{r,i,0} \right\} \quad (13)$$

where, variable c_l denote the amplitude of band pass filter and l varies from 0 to n. The output of equation (13) is passed through the logarithmic Mel- scaled filter bank to obtain Mel filter passed spectrum $f_{r,l,2}$

$$M_{FB}\ (f_{r,l,2}) = \left\{ 2597 * log_{10}(1 + f_{r,l,1}/700) \right\} \quad (14)$$

Discrete Cosine Transform (DCT) performed on equation (14) to get 13 coefficients of MFCC for each frame as:

$$f_{r,l,3} = \left\{ \sum_{k=1}^{N_d} \left(cos\left[\frac{i(2l-1)\pi}{2N_d}\right]\right)(f_{r,l,2}) \right\} \quad (15)$$

where k varies from 0 to $N_c > N_d$ and N_c, number of cepstral features used for further calculation.MFCC extracts 13 features in total out of which 12 are coefficients and 1 is energy feature. First delta $\triangle$ and double delta $\triangle\ \triangle$ feature also added to capture non−uniform behaviour of audio signal.We use the first 13 coefficients for our system, as the lower order coefficients provide more information about the source's overall spectral

structure[41].

2) *PLP*

PLP uses windowing and FFT, which are identical to MFCC and GTCC's first two phases. The Bark Filter Bank technique is then applied to the calculated frequency value. A filter bank with 27 highly sharp bandpass filters is included in Bark Filter. The Bark frequency with respect to a speech signal is given as:

$$f_{r,l,2} = \left\{ 6ln\left[\frac{f_{r,l,1}}{1200\pi} + \left[\left(\frac{f_{r,l,1}}{1200\pi}\right)^2 + 1\right]^{0.5}\right] \right\} \quad (16)$$

The Bark frequency component obtained is employed in the equal loudness emphasis step's pre-emphasis procedure. The following is the relationship between the discrete input power spectrum $f_{r,k,1}(k)$ and the LP model power spectrum $f_{r,l,2}(k)$.

$$\left\{ \frac{1}{n} \sum_{k=1}^{K-1} \left(\left[\frac{f_{r,l,2}(k)}{\hat{y}_{r,l,2}(k)}\right]\right) \right\} = 1 \quad (17)$$

After the Linear prediction, recursive cepstrum computation is applied to get the Perceptual Linear Prediction (PLP) coefficients. The first 13 coefficients are obtained, and then by using delta () and double delta () features, PLP feature vector is obtained.

3) *CQCC*

CQCC features are based on Constant Q Transform (CQT). In recent years, CQT has been used widely to analyse and classification of the audio signal. For extraction of CQCC, firstly, CQT is computed using equation (18) for discrete time-domain signal $x(i)$.

$$C^{Q,T}(K, N) = \left\{ \sum_{i=N-\lfloor N_i/2 \rfloor}^{N+\lfloor N_i/2 \rfloor} x(i)\, f_K{}^*\, (i - N + N_i/2) \right\} \quad (18)$$

where, K is the frequency bin index, f_K the complex conjugate of $F_K(N)$ and N_i is window length. In the second step, the cepstrum in a time sequence $x(i)$ is derived in spectrum logarithm using inverse transformation. However, cepstral features cannot be used directly because K bins of $C^{Q,T}$ are on a different scale than the cosine function of the DCT. To resolve this problem uniform resampling is done on cepstral features. In the final step, equation (19) is used to compute CQCC features.

$$C_{QCC}(\rho) = \left\{ \sum_{r=1}^{R} log\left|C^{Q,T}(r)\right|^2 cos\left[\frac{\rho\left(r-\frac{1}{2}\right)\pi/2}{R}\right] \right\} \quad (19)$$

where ρ= 0, 1, ..., L-1 and l are the newly resampled frequency bins.

4) GTCC

The only difference between MFCC and GTCC is that GTCC is more robust to noise[42]. The GTCC uses Gammatone filters with equivalent rectangular bandwidth (ERB) bands. To perform GTCC feature calculation, first FFT applied to the audio signal to generate a spectrum. To compute the energy E_n of each signal, gammatone filter bank applied to FFT audio signal. Then, the logarithm of each energy E_N is computed, and DCT is applied on signal to generate GTCC features. The function to compute GTCC feature is given as:

$$GTCC_L = \left\{ \sqrt{\frac{2}{g}} \sum_{g=1}^{G} logE_g cos\left[\frac{\pi g}{G} \left(l - \frac{1}{2}\right)\right] \right\} \quad (20)$$

where $1 \leq l \leq L$.

The signal energy for the nth spectral band is represented by E_g, the number of gammatone filters G, and the number of GTCC are represented by L. The 13-dimensional GTCC coefficients are returned by the GTCC computing method.

*5) **BFCC***

For addressing noisy speech signals, BFCC is thought to be a more powerful parameterization technique. Instead of utilizing Fourier transform-based Gammatone and Mel-scale filter-banks, BFCC employs auditory transform-based Gammachirp filter-bank. It employs basilar membrane functions and is more robust to additive noise than MFCC and GTCC. BFCC uses Equivalent Rectangular Bandwidth (ERB) function to calculate bandwidth and Cochlear Wavelet Transformation to form an auditory spectrogram. The equation to calculate gammachirp filter is defined as:

$$X = \left\{ xT^{N-1}e^{-2\pi BT} \right\} \quad (21)$$

$$g_{ch}(F,T) = \left\{ X * cos(2\pi FT + c_h logT + \varphi) \right\} \quad (22)$$

where, c_h is gammachirp function, and is the only difference between GTCC and BFCC. In last step, logarithmic and DCT operation applied to obtain BFCC features.

C. *Acoustic Models*

Some traditional ML methods are generative, while others are discriminative. These methods are suited for imposter detection in applicable datasets from ASV systems' initial research[43]. The proposed work applies multiple classification algorithms such as SVM, NB, DT and KNN. Also, deep learning algorithm such as LSTM based acoustic model has also been used to implement the acoustic model in the proposed work.

*1) **SVM***

SVM uses 2-D hyperplane to classify data into two classes, each of which is located on opposite sides of the plane[44], [45]. It has been effectively utilized for both speaker verification and spoof detection. However, due to the complex feature distribution, the different classes may be overlapped or intertwined, so the audio classes cannot be separated linearly [46]. A kernel-based SVM, on the other hand, is ideally adapted to handle such circumstances. Although, choosing a suitable kernel is critical for SVM classification accuracy. The following equation represents the ultimate optimal hyperplane classifier:

$$F(v) = \left\{ \sum_{l=1}^{L} \bar{c_l} \;\; w_i v_i v + \;\; \bar{C} \right\} \quad (23)$$

where, c, v represents classifier and support vector for solution, respectively. The equation (21) is updated for kernel SVM as:

$$F(v) = \left\{ \sum_{l=1}^{L} \bar{c_l} \;\; w_i v_i v + \;\; \bar{C} \right\} \quad (24)$$

The works proposed in [47], [48] show that SVM is highly good at handling noisy and high-dimensional data, and also, it achieves excellent accuracy with a small data set.

*2) **KNN***

KNN is a supervised learning algorithm in which a new instance is classified based on the feature space's closest training samples[49]. The Euclidean distance between the new instance and each previously-stored training audio clip is calculated to categorize a new one. A new audio clip is assigned to the most popular class among the K-training audio clips nearby [50]. For identifying the unlabelled data majority voting technique is employed [51]. It means that each occurrence of a class (category) in a set of K neighbourhood samples receives one vote. The fresh data sample is then classified into the class having the most votes. Because it is less susceptible to outliers, majority voting is more widely utilized. However, KNN classifier takes more computation time than SVM.

*3) **Naïve Bayes(NB)***

NB uses Bayes theorem to calculate to the probability of audio features with respect to class. Once trained, the NB classifier can predict emotions that aren't in the dataset [52]. In the proposed approach, two variations of NB have been used, Gaussian NB and Kernel NB. Following equation is used to calculate probability concerning class for Gaussian and Kernel NB.

$$P((X_i|Y)) = \left\{ \frac{1}{\sqrt{2\pi\sigma_Y{}^2}} e^{\left(-\frac{(X_i-\mu_Y)^2}{2\sigma_Y{}^2}\right)} \right\} \quad (25)$$

*4) **Decision Tree (DT)***

Decision Tree (DT): Classifiers that describe their classification information in the form of a tree are known as DT. A DTs' interior nodes represent a test on an attribute. If the test is passed, the instance being categorized will take one of the node's branches; if the test is not passed, the instance will take the other branch. Starting at the root node of the decision tree, an instance is classified by following a path given by attribute tests until it reaches a leaf node. A categorization or conclusion is represented by each leaf

node in a decision tree. In [53], SVM has been proven to be less resilient against label noise than decision tree. In the proposed approach multiple DT have been considered with distinct splitting node.

*5) **LSTM***

LSTM has the ability to learn long-term dependencies in data. Its architecture consists of a cell state and three gates that enable it to selectively learn, forget or retain information from each unit. By allowing only a few linear interactions, the cell state can carry information across the units without modifying it. Each unit has an input gate, an output gate, and a forget gate that adds or removes data from the cell state. LSTM uses following equation :

$$M_T = \left\{ b_F + \alpha\left(W_F * [h_{T-1}, x_T]\right) \right\} \tag{26}$$

$$I_T = \left\{ b_I + \alpha\left(W_I * [h_{T-1}, x_T]\right) \right\} \tag{27}$$

$$C_T = \left\{ b_C + tanh(W_C * [h_{T-1}, x_T]) \right\} \tag{28}$$

$$C_T = \left\{ C_T * M_T * C_{T-1} + i_T \right\} \tag{29}$$

$$O_T = \left\{ b_O + \alpha\left(w_O\,[h_{T-1}, x_T]\right) \right\} \tag{30}$$

$$h_T = \left\{ tanh() * O_T \right\} \tag{31}$$

where M_F, h_{T-1}, x_T and b_F represents weighted matrix, previously hidden state, input to the current state and weight associated with the information, respectively. In equation (25) M_I represents weight matrix associated with hidden state and b_I weight matrix of input. C_T in equation (27) represents a function of long-term memory and C_{T-1} is the cell state at current time. Equation (29) uses O_T and tanh to calculate current hidden state H_T. Kons et al. [54] used LSTM for Urban Sound classification and proved that LSTM works better than ML traditional algorithm.

3. Proposed Approach

As described earlier, front-end FE and back-end acoustic modelling are the two key components of the suggested ASV system. Figure 2 shows the proposed ASV system is implemented in two steps. Firstly, sequential integration of audio features with image features is carried out, and in secondly, LSTM and different ML algorithm-based acoustic models use these integrated features at back-end for classification. Ten pair and five trio feature sets have been generated using two image features LBP, ATP and five audio feature MFCC, PLP, CQCC, GTCC, and BFCC. Extracted features, along with the labels from the dataset, are then applied to the classification models i.e. Deep learning based model (LSTM), and ML based models (SVM, KNN, NB, and DT) used for assessing their performance. Table 1 shows the list of abbreviations used in this paper.

*1) **Sequential Integration of Audio and Image Features***

A total of fifteen feature sets have been created by gradually integrating two image features with five audio features. For instance, when ATP is sequentially integrated

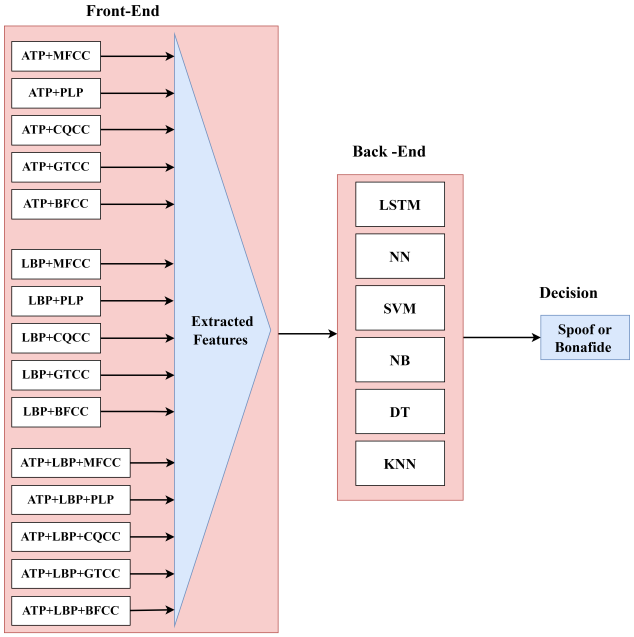


Figure 2. Proposed System Architecture

with MFCC, a set of ATP-MFCC feature vectors is created. Similarly, fifteen different combinations (ten pairs features and five trio features) are created such as ATP-MFCC, ATP-PLP, ATP-CQCC, ATP-BFCC, ATP-GTCC, LBP-MFCC, LBP-PLP, LBP-CQCC, LBP-BFCC, LBP-GTCC, ATP-LBP-MFCC, ATP-LBP-PLP, ATP-LBP-CQCC, ATP-LBP-BFCC, and ATP-LBP-GTCC. Algorithm 1 describes this process of sequential integration.

- **ATP and Audio Feature Integration:** : Figure 3 shows the process of creating pair of features by integrating image feature ATP with audio feature extraction. For feature extraction, audio signal from the dataset has been applied.By merging 20 ATP coefficients and 13 MFCC features, an integrated ATP-MFCC feature vector with 33 features is generated. Figure 3(a) shows steps involved in calculating ATP-MFCC features. Figure 3(b) shows the steps involved in calculating ATP-PLP features. An integrated ATP-PLP feature vector with 34 features is created by combining 20 ATP coefficients and 14 PLP features. As a result of this sequential combination of features, there are a total of 34 features. An integrated ATP-CQCC feature vector with 1565 features is constructed by combining 20 ATP features and 1545 CQCC coefficients. The procedure involved in calculating ATP-CQCC characteristics is depicted in Figure 3(c). Figure 3(d) shows steps involved in the process of sequentially combined ATP-GTCC feature. A feature vector of 33 features is constructed by combining 20 ATP features, and 13 GTCC coefficients. A 33-feature integrated ATP-BFCC feature vector is created by sequentially combining 20 ATP features and 13

TABLE I. Abbreviation

Abbreviation	Meaning
ASV	Automatic Speaker Verification
ATP	Acoustic Ternary Pattern
BFCC	Basilar-membrane Frequency-band Cepstral Coefficients
CRNN	Convolutional Recurrent Neural Network
CQCC	Constant Q Cepstral Coefficients
CNN	Convolutional Neural Network
DT	Decision Tree
ECOC	Error-Correcting Output Codes
FE	Feature Extraction
GMM	Gaussian Mixture Model
GTCC	Gammatone Cepstral Coefficients
HMM	Hidden Markov Model
IMFCC	Inverted Mel frequency cepstral coefficient
KNN	K- Nearest Neighbour
LBP	Local Binary Pattern
LSTM	Long Short-Term Memory
LDA	Linear Discriminant Analysis
LPCC	Linear Prediction Cepstral Coefficients
MFCC	Mel Frequency Cepstral Coefficients
ML	Machine Learning
NB	Naïve Bayes
NN	Neural Network
PLP	Perceptual Linear Prediction
SVM	Support Vector Machine

BFCC coefficients. Figure 3(e) shows steps involved in combining ATP-BFCC features.

- **LBP and Audio Feature Integration:** Figure 4 shows the process model of creating feature sets by combining LBP and Audio features. A Sequentially integrated LBP-MFCC feature vector of 61 features are produced by combining 13 MFCC coefficients and 48 LBP features. The procedure in calculating LBP-MFCC characteristics is depicted in Figure 4(a). Figure 4(b) shows steps involved in calculating LBP-PLP features,14 features from PLP and 48 features taken from LBP to create a sequentially combined feature set of 62. By sequentially merging 1545 CQCC coefficients and 48 LBP features, an integrated LBP-CQCC feature vector with 1593 elements is generated. Figure (c) shows, steps involved in calculating LBP-CQCC features. Figure 4(d) shows the process of integration. 13 features from GTCC and 48 features from LBP combined sequentially to create 61 feature set. A 61D feature set is created by sequentially combining 13 features from BFCC and 48 elements from LBP. The procedures done in calculating LBP-BFCC characteristics are depicted in Figure 4(e).

- **ATP-LBP and Audio Feature Integration:** Figure 5 shows the process of creating different feature set by combining ATP-LBP and audio features.An aggregated ATP-LBP-MFCC feature vector of 81 features is constructed by combining 20 ATP, 48 LBP features with13 MFCC coefficients. The procedures involved in calculating feature set are depicted in Figure 5(a). By merging 20 ATP, 48 LBP features and 14 PLP coefficients, an integrated ATP-LBP-PLP feature vector with 82 features is generated. Figure 5(b) shows the steps involved in calculating LBP-PLP characteristics. By merging 20 ATP, 48 LBP features and 1545 CQCC coefficients and, an integrated ATP-LBP-CQCC feature vector with 1613 features is generated. Figure 5(c) shows the steps involved in calculating feature set. By merging 20 ATP, 48 LBP feature with 13 GTCC coefficients, an integrated ATP-LBP-GTCC feature vector with 81 features is generated. Figure 5(d) shows, steps involved in calculating respective features set. An integrated ATP-LBP-BFCC feature vector of 81 features is constructed by integrating 20 ATP, 48 LBP features, and 13 BFCC coefficients. As a result of this sequential combination of features, there are a total of 33 features. The procedures needed in calculating feature set are shown in Figure

*2) **Back-end Classification Models***

In the proposed approach, five different classification models are designed using four different ML algorithms, that are, NB, SVM, DT and KNN, and one deep learning algorithm i.e. LSTM. ASV Spoof 2019 LA training partition has been used to train the model and for testing purpose ASV Spoof 2019 evaluation partition has been used. Figure

Algorithm 1 Classifying audio into Spoofed or bonafide

Input: Audio wave file in FLAC form

Begin:

Feature Extraction Computation

Select type of image feature extraction technique.

If ($feature == ATP \parallel feature == LBP$)

A ← ATP_2D [ ]

Else

$B \leftarrow LBP\_48[]$

End if

Select audio feature extraction technique

If($feature == MFCC \| feature == BFCC \| feature == PLP \| feature == GTCC \| feature == CQCC$)

M← MFCC_13D []

P← PLP_14D []

C← CQCC_1545D []

G← GTCC_13D[]

B← BFCC_13D[]

End if

Sequential Integration of feature:

$AM = Concatenate(A, M)$

$AP = Concatenate(A, P)$

$AB = Concatenate(A, B)$

$AG = Concatenate(A, G)$

$AC = Concatenate(A, C)$

$LM = Concatenate(L, M)$

$LP = Concatination(LP)$

$LB = Concatenate(L, B)$

$LG = Concatenate(L, G)$

$LC = Concatenate(L, C)$

$ALM = Concatenate(A, L, M)$

$ALP = Concatenate(A, L, P)$

$ALB = Concatenate(A, L, B)$

$ALG = Concatenate(A, L, G)$

$ALC = Concatenate(A, L, C)$

Classification :

If ($feature\ set = AM \parallel feature\ set = AP \parallel feature\ set = AB \parallel feature\ set = AG \parallel feature\ set = AC \parallel feature\ set = LM \parallel feature\ set = LP \parallel feature\ set = LB \parallel feature\ set = LG \parallel feature\ set = LC \parallel feature\ set = ALM \parallel feature\ set = ALP \parallel feature\ set = ALB \parallel feature\ set = ALG \parallel feature\ set = ALC$)

LSTM (feature set)

NN (feature set)

SVM (feature set)

KNN (feature set)

NB (featureset)

DT (featureset)

End if

End

6 shows Process of Training and Evaluation of Proposed Work.

- **Neural Network:**In our proposed approach, five different NN has been built by changing the number of layers. The first NN, called Narrow Neural Network(N), has been designed with one fully connected layer containing one layer with ten neurons, and as an activation function, relu has been used. Medium NN(M), designed with one fully connected layer and include one layer with 1025 neurons second. The third NN, called Wide Neural Network(W), has been created, which contains one fully connected and the first layers built using 100 neurons. The fourth NN, called Bi- Neural Network(B), has been developed, which includes two fully connected layer and the

TABLE II. Performance of Integrated ATP and Audio Features using NN

Feature set	Algorithm	Accuracy (%)	Precision (%)	Recall (%)	F1score (%)	EER (%)
ATP+MFCC	N_NN	95.8	81	75	78	21
	M_NN	95.7	78	79	78	25
	W_NN	96.1	96	96	95.8	12
	B_NN	96	95	94	94.9	13
	T_NN	95	78	75	76	23
ATP+PLP	N_NN	90.2	55	18	27	46
	M_NN	85.5	31	26	29	47
	W_NN	84.9	26	26	26	48
	b_NN	89.5	46	20	28	47
	T_NN	89.2	43	21	28	45
ATP+CQCC	N_NN	87.8	43.5	43.5	43.5	44
	M_NN	88.4	45.8	40.7	43.1	41
	W_NN	89	48.9	43.5	46	45
	B_NN	87.9	43	38.4	40.5	42
	T_NN	88.5	46.5	43.5	44.9	46
ATP+GTCC	N_NN	96.1	96	96	96	10
	M_NN	95.6	78	77	78	34
	W_NN	95	78	77	78.9	37
	b_NN	95.6	95	94	94	12.5
	T_NN	95.6	80	76	78	35
ATP+BFCC	N_NN	95.7	80	75	78	29
	M_NN	95.3	77	77.1	77.2	33
	W_NN	95.6	78.5	77	78	34
	W_NN	96.2	82	79	81	28
	T_NN	96.7	79	77	78	30

TABLE III. Performance of Integrated LBP and Audio Features using NN

Feature set	Algorithm	Accuracy (%)	Precision (%)	Recall (%)	F1score (%)	EER (%)
LBP+MFCC	N_NN	91.1	54.4	55.5	55	41
	M_NN	92	60	51	55	39
	W_NN	92.4	62.5	56	59	36
	B_NN	91.8	59	53	57	44
	T_NN	91.3	61.2	52.2	56.4	41
LBP+PLP	N_NN	92.9	63.9	62	63.2	39
	M_NN	93.1	66.2	59.5	62.7	37
	W_NN	93.3	68.6	57.5	62.6	36
	b_NN	91.9	54.8	51.5	53.1	46
	T_NN	90.6	52	52.5	52.2	43
LBP+CQCC	N_NN	88.8	41.2	33	36	47
	M_NN	90.0	48.6	36.3	41.6	45
	W_NN	90.7	53.1	42.4	47.1	46
	B_NN	89.6	46.5	41.4	43.8	42
	T_NN	91.1	56.5	39.3	46.4	40
LBP+GTCC	N_NN	91.5	56.4	57.5	57	48
	M_NN	92.9	64.5	60.6	62.5	36
	W_NN	91.9	58.5	58.5	58.5	45
	b_NN	92.2	61.1	55.5	58.2	41
	T_NN	92.5	63.5	54.5	58.6	40
LBP+BFCC	N_NN	93.1	66.6	58.5	62.3	40
	M_NN	93.6	69.3	61.6	65.2	37
	W_NN	93.7	71.6	58.5	64.4	35
	b_NN	91.3	55.9	52.5	54.1	45
	T_NN	92.2	60.4	58.5	59.4	42

TABLE IV. Performance of Integrated ATP, LBP and Audio Features using NN

Feature set	Algorithm	Accuracy (%)	Precision (%)	Recall (%)	F1score (%)	EER (%)
ATP+LBP+MFCC	N_NN	91.8	90.5	90.5	91.6	27
	M_NN	90.3	50.6	42.4	46.5	45
	W_NN	91.8	59	52.5	55.6	41
	B_NN	97.7	96.8	96.4	74.4	15
	T_NN	89.1	44.3	43.4	43.8	43
ATP+LBP+PLP	N_NN	90.9	53.9	48.4	51	45
	M_NN	91.6	57.4	54.5	55.9	44
	W_NN	92.7	64.7	55.5	59.7	36
	b_NN	89.6	46.2	37.3	41.3	47
	T_NN	90.9	54.1	46.4	50	43
ATP+LBP+CQCC	N_NN	86.6	47	40	43	41
	M_NN	89.7	47	43	45	45
	W_NN	89.5	46	41	44	43
	B_NN	88.2	40	38	39	44
	T_NN	88.1	40	41	40	40
ATP+LBP+GTCC	N_NN	92.6	64	57	60	35
	M_NN	91.4	56	56	56	48
	W_NN	92.4	63	56	59	36
	b_NN	91.8	91	91	91.6	12
	T_NN	90.8	53	54	53	42
ATP+LBP+BFCC	N_NN	91.8	50.5	50.5	54.6	37
	M_NN	92.3	60.3	62	61	40
	W_NN	92.3	60	62	61	44
	b_NN	91.6	59	48	53	47
	T_NN	90.6	52	43	48	43

$^{*}N\_NN : NarrowNeuralNetwork$, $^{*}M\_NN : MediumNeuralNetwork$, $^{*}W\_NN : WideNeuralNetwork$, $^{*}B\_NN : Bi-NeuralNetwork$, $^{*}T\_NN : Tri-NeuralNetwork$

first layers made using ten neurons, and the second layer also contains10 neurons. Fifth, NN called Tri-Neural Network(T), has been designed using three fully connected and its first layers, second layers, and third layer contains10 neurons each.

- **Naïve Bayes:** In our approach, two variations of NB have been used to classify the dataset. Gaussian NB uses a Gaussian normal distribution that works on a continuous feature dataset. The kernel-based NB(K) algorithm gives good accuracy when data is not linearly separable.

- **Support Vector Machine:** SVM is a Supervised classification algorithm. The SVM algorithms' purpose is to find the optimum line or decision boundary for categorizing n-dimensional space into classes so that additional data points can be readily placed in the correct category in the future. A hyperplane is the optimal choice boundary. When data is linearly separable, it's easy to classify the new data point, but when information is not linearly separable, there need to modify SVM. Different kernel function has been used for this process. In the proposed approach, four kernel functions have been used: Linear(L), Quadratic(Q), Cubic(C), and Gaussian(G) function.

- **Decision tree:** Decision Trees are a type of Supervised ML algorithm. In DT, the data is continually split according to some parameter. Two entities, decision nodes and leaves, can be used to explain the tree. The leaves represent the decisions or outcomes. The decision tree created at different levels by setting the number of splits. Gini index has been used to calculate the probability of a specific feature classified incorrectly when selected randomly. The result of these experiments is explained in the next section. In proposed approach, Decision trees created at three levels for evaluation: Coarse-level(Coa), with only a few decision nodes (maximum number of splits is four); Medium-level(Me), with more decision nodes (maximum number of splits is twenty); and Fine-level(F), with a large number of decision nodes (maximum number of splits is one hundred). Fine trees have greater depth in their structure, while coarse trees have the least.

- **K- Nearest Neighbour:** For audio classification, the KNN model has been used. KNN trained on a dataset obtained from the audio signal at the front end. KNN finds the nearest neighbour by calculating the distance between data points. In our proposed approach, three parameters have been used to tune

TABLE V. Performance of Integrated ATP and Audio Features using SVM

Feature set	Algorithm	Accuracy (%)	Precision (%)	Recall (%)	F1score (%)	EER (%)
ATP+MFCC	L_SVM	89.7	50	10	19	50
	Q_SVM	90.1	81	65	72	18
	C_SVM	86.7	83	79	81	11.2
	G_SVM	90.9	89	48	62	49.9
ATP+PLP	L_SVM	89.8	38	18	24	50
	Q_SVM	89.8	35	17	23	50
	C_SVM	89	42	21	28	41
	G_SVM	89.8	40	21	20	50
ATP+CQCC	L_SVM	89.2	45	18	13	50
	Q_SVM	89.2	55	17	53	37.8
	C_SVM	80.8	52	21	51	50
	G_SVM	89.2	50	40	55	48.2
ATP+GTCC	L_SVM	94.5	94	94	94.4	2.3
	Q_SVM	94.5	91.5	92.9	94.4	2.1
	C_SVM	98.3	98	98.5	98.7	1.4
	G_SVM	93.9	88.3	85.5	88.1	27.5
ATP+BFCC	L_SVM	89.8	75.3	70.5	75.7	50
	Q_SVM	90.6	80.5	62.4	70	19.6
	C_SVM	95.3	79.3	72.5	75.7	14.6
	G_SVM	84.6	90	46	61	26.9

KNN experiments: the number of neighbours (KN), the distance metric used to determine the nearest neighbours, and the distance weights. For the first three tests, different values of neighbours and set the distance metric to Euclidean and distance weight to equal; initially KN=1,fine KNN(F_KK), then KN =10,medium KNN(Me_KNN), and finally KN =100, Coarse KNN(Co_KNN). In the second experiment, the number of neighbours (KN=10) and distance weight are equal while altering the distance metric to cosine(Cos_KNN) and cubic(Cu_KNN). Each NN is given a weight based on the squared inverse mechanism, assigning higher weights to closer neighbours and lower weights to farther neighbours.

- **LSTM**: The proposed LSTM network is made up of 50 LSTM layers, each with a relu activation function. Audio and other time-varying data frames are particularly well suited to LSTM layers. The suggested LSTM model has 50, 50, and 50 units in the LSTM layers, and one dense output layer in this configuration. The effect of some randomly chosen neurons is then turned off using a 20 per cent dropout. The addition of a dropout layer prevents overfitting in the model. After that, the dropout outcome is transmitted to a dense layer with a sigmoid activation function. LSTM trained by feature matrix, which has been produced at the front end separately.

4. Experimental Setup and Result

This section covers experimental details of the proposed ASV system. Feature extraction implemented on MATLAB R2021 and Windows10 Operating System with intel core i5 processor has been used for processing. The inbuilt $mfcc()$ and $gtcc()$ in MATLAB have been used to extract MFCC and GTCC spectrogram, respectively. Anaconda has been used to implement the back-end model written in Python 3.7. Back-end model implemented on Anaconda using python 3.7. All audios and labels acquired from the ASVspoof2019 training, evaluation datasets. For ML approaches, MATLAB inbuilt classification application has been used. Different parameters have been used to measure the performance of the proposed ASV system such as Accuracy , Precision , Recall , F1-score , and EER .

1) Performance of different Feature Integrations with Neural Network-based Acoustic Model

This section presents the results obtained using various proposed front end feature combinations with Neural Network-based acoustic model at the back-end. As described earlier, five different NN has been built by changing the number of layers such as Narrow NN, Medium NN, Wide NN, Bi-layered NN and Tri-layered NN.

- **Performance of Integrated ATP and Audio Features:** Table 2 gives the results for the proposed ASV system that uses ATP features integrated with audio features at front end and different types of Neural Network based acoustic model at back-end. It can clearly be observed from the results that ATP-MFCC feature set produces best EER 12% using Wide NN compared to Narrow, Medium, Bi-layered, and Tri-layered NN. ATP-PLP achieved the best EER 45 % with Tri-layered NN compared to other NN. ATP-CQCC achieved 44% EER with Narrow NN, ATP-GTCC and ATP-BFCC gained 27%, 28% using Narrow NN and Wide NN, respectively. Hence, it can

TABLE VI. Performance of Integrated ATP and Audio Features using SVM

Feature set	Algorithm	Accuracy (%)	Precision (%)	Recall (%)	F1score (%)	EER (%)
LBP+MFCC	L_SVM	90.2	50	1	1	50
	Q_SVM	93.7	81	45	58	27.8
	C_SVM	92.9	65	56	60	23.3
	G_SVM	90.8	1	4	7.7	48
LBP+PLP	L_SVM	90.9	50	1	1	50
	Q_SVM	93.4	82	41	55	29.8
	C_SVM	92.8	67	51	58	25.6
	G_SVM	90.9	1	7	1.3	46.4
LBP+CQCC	L_SVM	90.2	50	1	1	50
	Q_SVM	90.5	49	25.2	33.3	38
	C_SVM	88.4	50	37.3	42.7	33.4
	G_SVM	90.2	66.6	20	39	40
LBP+GTCC	L_SVM	90.2	50	1	1.9	50
	Q_SVM	93.8	70	33	45	26.08
	C_SVM	92.6	60	48.4	53.9	22.5
	G_SVM	91.2	50	1	1.9	44.9
LBP+BFCC	L_SVM	90	50	1	1	50
	Q_SVM	94.5	84	55	66	23.05
	C_SVM	94	68	75	71.4	14
	G_SVM	90.5	1	7	1.3	47.5

TABLE VII. Performance of Integrated ATP and Audio Features using SVM

Feature set	Algorithm(SVM)	Accuracy (%)	Precision (%)	Recall (%)	F1score (%)	EER (%)
ATP+LBP+MFCC	L_SVM	92.2	50	1	1	50
	Q_SVM	92.5	72	37	49.3	32.4
	C_SVM	91.4	57	45	50.8	29.05
	G_SVM	90.2	50	1	19	50
ATP+LBP+PLP	L_SVM	92.2	50	1	1	50
	Q_SVM	92	66	37	47	32.3
	C_SVM	91.9	61	46.4	52.8	28.3
	G_SVM	90.2	50	1	19	50
ATP+LBP+CQCC	L_SVM	90.2	50	1	1	50
	Q_SVM	90.1	49	25.2	33.3	38.8
	C_SVM	90.2	50	37.3	42.7	33.3
	G_SVM	90.4	66.6	20	39	49
ATP+LBP+GTCC	L_SVM	96	96	96	96	2.6
	Q_SVM	95.5	95.5	92.9	95.4	2
	C_SVM	99.3	99	98.5	98.7	1
	G_SVM	98.9	98.3	95.5	98.1	2.5
ATP+LBP+BFCC	L_SVM	90.2	50	1	19	50
	Q_SVM	93.2	75	44.4	56	28.5
	C_SVM	93.2	75	44.4	56	28.5
	G_SVM	90.2	50	1	1	50

$^{*}L\_SVM$: $LinearSupportVectorMachine$, $^{*}Q\_SVM$: $QuadraticSupportVectorMachine$,
$^{*}C\_SVM$: $CubicSupportVectorMachine$, $^{*}G\_SVM$: $GaussianSupportVectorMachine$

TABLE VIII. Performance of Integrated ATP and Audio Features using KNN

Feature set	Algorithm	Accuracy (%)	Precision (%)	Recall (%)	F1score (%)	EER (%)
ATP+MFCC	F_K	93.3	93	93	92	6.9
	Me_K	92.8	81	37	56.9	31.9
	Co_K	90	91	18	33	49.1
	Cos_K	92.1	80	28	42	35.6
	Cu_K	92.9	82	37	51	31.1
	We_K	92.8	87	33	48	33.2
ATP+PLP	F_K	84	50	55	54	41.9
	Me_K	90.1	58	8	14	45.9
	Co_K	89.8	50	4	11	50
	Cos_K	89.9	51	6	11	46.9
	Cu_K	90.2	63	8	15	45.7
	W_K	90.3	67	7	14	46.1
ATP+CQCC	F_K	87.7	42	36	39	31.6
	Me_K	89	46	13	20	43.3
	Co_K	89.2	50	46	50	42
	Cos_K	89.4	51	18	27	40.5
	Cu_K	89.1	47	11	18	44.2
	We_K	89.2	49	11	18	44
ATP+GTCC	F_K	94.9	94	94	94	6.8
	Me_K	92.3	77	34	47	32.8
	Co_K	90	90	13	27	49.3
	Cos_K	92.1	84	27	41	36.3
	Cu_K	92.4	78	35	48	32.4
	We_K	92.4	83	30	45	34.5
ATP+BFCC	F_K	91	56.1	50	52	27.15
	M_K	92.4	78	34	48	33.1
	Co_K	90.1	95	31	60	48.4
	Cos_K	91.9	82	25	39	37.2
	Cu_K	92.6	80	35	49	32.1
	We_K	92.3	82	31	45	34.5

be concluded that ATP-GTCC outperforms all other feature set combinations with NN based acoustic model at back-end.

- **Performance of Integrated LBP and Audio Features:** Table 3 gives the results for the proposed ASV system that uses LBP features integrated with audio features at front end and different types of Neural Network based acoustic model at back-end. In is clearly observed from table that LBP-MFCC feature set produces best EER 36% using Wide Neural network. LBP-PLP achieved the best EER 36% result with Wide NN. LBP-CQCC and LBP-GTCC achieved 40% and 36% using Medium NN and Tri-layered NN, LBP-BFCC achieved 35% EER with Wide NN. Hence, it can be concluded that LBP-BFCC outperforms all other feature set combinations with NN based acoustic model at back-end.

- **Performance of Integrated ATP, LBP, and Audio Features:** Table 4 gives the result for the proposed ASV system that uses ATP-LBP features integrated with audio features at front end and different types of Neural Network based acoustic model at back-end. It is clearly observed from table that ATP-LBP-MFCC feature set produces best EER 15% using bi-layered Neural network. ATP-LBP-PLP achieved the best EER 36% result with Wide NN. ATP-LBP-CQCC gained 41% using Narrow NN, ATP-LBP-GTCC gained 1.5% with bi-layered NN, ATP-LBP-BFCC achieved 37% EER with Narrow NN. Hence, it is concluded that ATP-LBP-GTCC outperforms all other feature set combinations with NN based acoustic model at back-end.

2) *Performance of different Feature Integrations with SVM-based Acoustic Model*

This section presents the results obtained using various proposed front end feature combinations with SVM acoustic model at the back-end. As described earlier, four different SVM model has been built by changing their kernel function such as Linear, Quadratic, Cubic kernel function, and Gaussian kernel function.

- **Performance of Integrated ATP and Audio Feature:** Table 5 gives the results for the proposed ASV

TABLE IX. Performance of Integrated LBP and Audio Features using KNN

Feature set	Algorithm	Accuracy (%)	Precision (%)	Recall (%)	F1score (%)	EER (%)
LBP+MFCC	F_K	91.2	57	38	46	30.8
	Me_K	91.4	83	15	25	42.4
	Co_K	90.2	50	30	35	47.5
	Cos_K	90.8	80	28	46	47
	Cu_K	90.9	76	20	27	45
	We_K	91.3	92	22	24	43
LBP+PLP	F_K	90.6	56	37	45	30
	Me_K	91.4	84	25	35	42.4
	Co_K	90.2	51	35	55	50
	Cos_K	90.4	82	38	56	48
	Cu_K	91	78	30	37	44
	We_K	91	92	32	54	45
LBP+CQCC	F_K	89.3	44	33	37	33.2
	Me_K	91	66	36	46	41
	Co_K	90.2	50	31	39	50
	Cos_K	89.8	44	36	43	40
	Cu_K	90.4	53	45	53	42
	We_K	91.1	71	45	55	39
LBP+GTCC	F_K	89.7	46	37	41	31.3
	Me_K	90.9	65	35	44	42.4
	Co_K	90.2	50	37	39	50
	Cos_K	90.5	83	50	55	47
	Cu_K	90.8	66	42	50	43
	We_K	91.3	78	25	35	42.4
LBP+BFCC	F_K	90.6	52	43	47	30.5
	Me_K	90.6	60	12	20	43.7
	Co_K	90.2	50	10	19	50
	Cos_K	90.6	83	50	60	45
	Cu_K	90.6	62	55	67	46
	We_K	90.8	68	51	59	44

system that uses ATP features integrated with audio features at front end and different types of SVM based acoustic model at back-end. In is clearly observed from table that ATP-MFCC, ATP-PLP, ATP-CQCC, ATP-GTCC, and ATP-BFCC integrated feature produces 11.2%, 41%, 37.85%, 1.4%, 14.6% EER using Cubic SVM, Quadratic SVM respectively. Hence, it is concluded that ATP-GTCC outperforms all the other feature set.

- **Performance of Integrated LBP and Audio Feature:** The performance of each feature set used at back-end is shown in Table 6. It is clearly observed from the table that LBP-MFCC, LBP-PLP, LBP-CQCC, LBP-GTCC, and LBP-BFCC integrated feature gives 23.3%, 25.6%, 33.4%, 22.5%, 14% EER respectively using cubic SVM. Hence, it is concluded that LBP-BFCC outperforms all the other feature set.

- **Performance of Integrated ATP, LBP, and Audio Feature:** Table 7 gives the results for the proposed ASV system that uses ATP-LBP features integrated with audio features at front end and different types of SVM based acoustic model at back-end. It is observed from table that using cubic SVM, the integrated features of ATP-LBP-MFCC, ATP-LBP-PLP, ATP-LBP-CQCC, ATP-LBP-GTCC, and ATP-LBP-BFCC, yield 29.05%, 28.3%, 33.3%, 27.4%, 1%, and 33.3% EER, respectively. Hence it is concluded that ATP-LBP-GTCC outperforms all the other feature set.

*3) **Performance of different Feature Integrations with KNN-based Acoustic Model***

This section presents the results obtained using various proposed front end feature combinations with KNN acoustic model at the back-end. As described earlier, six different KNN model has been built by changing distance function and value of K such as Fine, Medium, Coarse, Cosine, Cubic, Weighted KNN.

- **Performance of Integrated ATP and Audio Feature:** Table 8 gives the results for the proposed ASV system that uses ATP features integrated with audio features at front end and different types of KNN based acoustic model at back-end. In is clearly observed from table that ATP-MFCC, ATP-PLP, ATP-CQCC, ATP-GTCC, ATP-BFCC integrated feature

TABLE X. Performance of Integrated ATP,LBP and Audio Features using KNN

Feature set	Algorithm	Accuracy (%)	Precision (%)	Recall (%)	F1score (%)	EER (%)
ATP+LBP+MFCC	F_K	94.3	93	92	93.5	6.7
	Me_K	90.9	73	31	39	44.3
	Co_K	90.2	50	31	39	50
	Cos_K	90.5	75	38	35	44
	Cu_K	90.7	72	70	74	45
	We_K	90.8	80	65	68	45.5
ATP+LBP+PLP	F_K	89.7	46	29	35	35
	Me_K	90.3	57	40	59	48
	Co_K	90.2	50	25	20	50
	Cos_K	90.2	50	30	35	48
	Cu_K	89.9	28	20	27	48
	We_K	90.1	50	20	30	50
ATP+LBP+CQCC	F_K	87.9	36	31	33	34
	Me_K	89.7	41	33	42	43
	Co_K	90.2	50	10	19	50
	Cos_K	89.9	45	17	25	41
	Cu_K	89.8	41	20	16	45
	We_K	90.2	50	12	19	43
ATP+LBP+GTCC	F_K	95.8	94	95	94	6.1
	Me_K	90.9	76	20	27	45
	Co_K	90.2	50	10	19	50
	Cos_K	90.4	75	30	58	97
	Cu_K	91.4	87	14	24	42
	We_K	91	76	10	17	49
ATP+LBP+BFCC	F_K	89.5	44	29	35	35
	Me_K	90.7	64	11	18	44
	Co_K	90.2	50	10	19	50
	Cos_K	90.3	66	20	39	49
	Cu_K	90.6	64	39	35	45
	We_K	90.9	76	30	37	44

$^{*}F\_K : FineK - nearestNeighbour$, $^{*}Me\_K : MediumK - nearestNeighbour$, $^{*}Co\_K : CoarseK - nearestNeighbour$, $^{*}Cos\_K : CosineK - nearestNeighbour$, $^{*}Cu\_K : CubicK - nearestNeighbour$,$^{*}We\_K : WeightedK - nearestNeighbour$

TABLE XI. Performance of Integrated ATP and Audio Features using NB

Feature set	Algorithm	Accuracy (%)	Precision (%)	Recall (%)	F1score (%)	EER (%)
ATP+MFCC	K_NB	85.8	85	85.5	86	26
ATP+PLP	K_NB	81.4	79	49	53	33.5
ATP+CQCC	K_NB	87.1	87	88	87.7	22
ATP+GTCC	K_NB	77.4	78	80	79	24
ATP+BFCC	K_NB	77.3	52	44	57	35

TABLE XII. Performance of Integrated LBP and Audio Features using NB

Feature set	Algorithm	Accuracy (%)	Precision (%)	Recall (%)	F1score (%)	EER (%)
LBP+MFCC	K_NB	78.3	66	61	73	39.5
LBP+PLP	K_NB	80.5	82	65	79	27
LBP+CQCC	K_NB	89.9	82	63	79	30
LBP+GTCC	K_NB	74.3	72	67	54	35
LBP+BFCC	K_NB	75	63	65	69	33.1

TABLE XIII. Performance of Integrated ATP,LBP and Audio Features using NB

Feature set	Algorithm	Accuracy (%)	Precision (%)	Recall (%)	F1score (%)	EER (%)
ATP+LBP+MFCC	K_NB	87	86	86	83	23.1
ATP+LBP+PLP	K_NB	79	63	45	53	37
ATP+LBP+CQCC	K_NB	90	87	88	87.5	20
ATP+LBP+GTCC	K_NB	77.1	65	49	59	32.1
ATP+LBP+BFCC	K_NB	75.2	60	43	58	44.2

*K_NB : $KernalNaiveBayes$

TABLE XIV. Performance of Integrated ATP and Audio Features using DT

Feature set	Algorithm	Accuracy (%)	Precision (%)	Recall (%)	F1score (%)	EER (%)
	F_DT	91.9	90	90.3	90	160
	Me_DT	90.9	89	89	88	25
ATP+MFCC	Coa_DT	89.8	60	55	59	39
	F_DT	88.1	25	9	13	45
	Me_DT	89.8	44	30	45	40
ATP+PLP	Coa_DT	89.8	50	29	53	50
	F_DT	86.1	34	31	32	34
	Me_DT	87.1	36	26	30	37
ATP+CQCC	Coa_DT	87.6	40	29	30	35
	F_DT	90.8	90	90	89	15
	Me_DT	90.1	89	86	90	19
ATP+GTCC	Coa_DT	89.8	89	87	84.5	22
	F_DT	90.2	52	44	48	27
	Me_DT	89.8	50	15	20	50
ATP+BFCC	Coa_DT	89.8	50	15	20	50

produces 6.9, 41.9%, 31.6%, 6.8%, 27.1% EER using Fine KNN. Hence it is concluded that ATP-GTCC outperforms other features.

- **Performance of Integrated LBP and Audio Feature:** Table 9 gives the results for the proposed ASV system that uses LBP features integrated with audio features at front end and different types of KNN based acoustic model at back-end. In is clearly observed from table that LBP-MFCC, LBP-PLP, LBP-CQCC, LBP-GTCC, LBP-BFCC and provides 30.8, 30, 33.2, 31.3, 30.5 % EER using fine KNN. Hence it is concluded that LBP-PLP outperforms other feature set.

- **Performance of Integrated ATP, LBP and Audio Feature:** Table 10 gives the results for the proposed ASV system that uses ATP-LBP features integrated with audio features at front end and different types of SVM based acoustic model at back-end. In is clearly observed from table that ATP-LBP-MFCC, ATP-LBP-PLP, ATP-LBP-CQCC, ATP-LBP-GTCC, and ATP-LBP-BFCC provides 6.7, 35, 34, 6.1, 35% EER respectively. Hence, it is concluded that ATP-LBP-GTCC surpass other feature sets.

*4) **Performance of different Feature Integrations with NB-based Acoustic Model***

This section presents the results obtained using various proposed front end feature combinations with NB acoustic model at the back-end. As described earlier, two different NB models have been built such as Gaussian NB and Kernel NB.

- **Performance of Integrated ATP and Audio Feature:** Table 11 gives the results for the proposed ASV system that uses ATP features integrated with audio features at front end and different types of NB based acoustic model at back-end. In is clearly observed from table that the kernel produced better accuracy with all fifteen-feature sets. Hence, it is concluded that ATP and MFCC integrated feature sets outperforms another feature sets.

- **Performance of Integrated LBP and Audio Feature:** Table 12 gives the results for the proposed ASV system that uses LBP features integrated with audio features at front end and different types of NB based acoustic model at back-end. In is clearly observed from table that LBP-PLP integrated feature set surpass other feature sets.

- **Performance of Integrated ATP, LBP, and Audio Feature**: Table 13 gives the results for the proposed ASV system that uses ATP-LBP features integrated

TABLE XV. Performance of Integrated LBP and Audio Features using DT

Feature set	Algorithm	Accuracy (%)	Precision (%)	Recall (%)	F1score (%)	EER (%)
	F_DT	89.9	88	84	86	25
	Me_DT	90.5	51	42	46	27
LBP+MFCC	Coa_DT	90	47	16	24	41
	F_DT	89.4	45	40	42	28
	Me_DT	88.8	40	29	33	35
LBP+PLP	Coa_DT	89.3	28	6	10	43
	F_DT	86.7	35	42	38	32.5
	Me_DT	86.7	38	40	39	33.4
LBP+CQCC	Coa_DT	89	43	39	41	33
	F_DT	88.4	40	37	38	34
	Me_DT	90.3	50	37	43	31
LBP+GTCC	Coa_DT	89.7	39	20	14	45
	F_DT	88.7	43	47	45	26
	Me_DT	89.5	46	44	45	24
LBP+BFCC	Coa_DT	90.3	51	18	26	40

TABLE XVI. Performance of Integrated ATP,LBP and Audio Features using DT

Feature set	Algorithm	Accuracy (%)	Precision (%)	Recall (%)	F1score (%)	EER (%)
	F_DT	92.5	92	92.3	92	16
	Me_DT	91	90	89	90	24
ATP+LBP+MFCC	Coa_DT	89	60	59	58	35
	F_DT	88.8	42	42	42.6	31.9
	Me_DT	89.7	42	34	38	32.7
ATP+LBP+PLP	Coa_DT	88.9	33	13	18	44.9
	F_DT	86.6	32	34	33	36.7
	Me_DT	87.9	37	33	35	36.4
ATP+LBP+CQCC	Coa_DT	90.8	55	29	38	36.6
	F_DT	93.8	93	92.5	92	13
	Me_DT	91	89	26	30	27
ATP+LBP+GTCC	Coa_DT	88.8	65	64	64.5	38
	F_DT	86.2	30	32	31	37.8
	Me_DT	87.2	35	38	37	34.5
ATP+LBP+BFCC	Coa_DT	89.2	38	17	23	42.9

$^{*}F\_DT : FineDecisionTree$, $^{*}Me\_DT : MediumDecisionTree$, $^{*}Coa\_DT : CoarseDecisionTree$

TABLE XVII. Performance of Integrated ATP and Audio Features using LSTM

Feature set	Accuracy (%)	Precision (%)	Recall (%)	F1score (%)	EER (%)
ATP+MFCC	98.5	98	97	98	2
ATP+PLP	95	91	98	95	1.2
ATP+CQCC	96.8	98.5	97.9	98.2	2
ATP+GTCC	98.5	97	98	97	1.1
ATP+BFCC	96	98	97	97	2.4

TABLE XVIII. Performance of Integrated LBP and Audio Features using LSTM

Feature set	Accuracy (%)	Precision (%)	Recall (%)	F1score (%)	EER (%)
LBP+MFCC	91	94	96	89	3.3
LBP+PLP	91	91	98	95	2.2
LBP+CQCC	92	94	97	90	2
LBP+GTCC	94	94	98	88	1.1
LBP+BFCC	91	93	96	89.1	3.9

TABLE XIX. Performance of Integrated ATP,LBP and Audio Features using LSTM

Feature set	Accuracy (%)	Precision (%)	Recall (%)	F1score (%)	EER (%)
ATP+LBP+MFCC	96	94	96	84	3
ATP+LBP+PLP	89	89	81	94	10
ATP+LBP+CQCC	89	89	82	84	10
ATP+LBP+GTCC	99.5	99	99.2	98	0.5
ATP+LBP+BFCC	90	91	98	94	9

TABLE XX. Performance of the Proposed ASV Systems

Feature set	Algorithm	Accuracy (%)	Precision (%)	Recall (%)	F1score (%)	EER (%)
ATP+MFCC	B_NN	96	95	94	94.9	13
ATP+GTCC	C_SVM	98.3	98	98.5	98.7	1.4
	F_K	94.9	94	94	94	6.8
	F_DT	90.8	90	90	89	15
	LSTM	90	91	98	94	1.1
ATP+CQCC	K_NB	87.1	87	88	87.7	22
LBP+PLP	F_K	90.6	56	37	45	30
LBP+GTCC	C_SVM	94	68	75	71.4	14
	LSTM	94	94	98	88	1.1
LBP+BFCC	W_NN	93.7	71.6	58.5	64.4	35
	Me_DT	89.5	46	44	45	24
ATP+LBP+CQCC	K_NB	90	87	88	87.5	20
ATP+LBP+GTCC	B_NN	91.8	91	91	91.6	12
	C_SVM	99.3	99	98.5	98.7	1
	F_KNN	95.8	94	95	94	6.1
	F_DT	93.8	93	92.5	92	13
	LSTM	96	98	97	98	0.5
ATP+LBP+BFCC	C_SVM	93.2	71	50	59	25.9

with audio features at front end and different types of NB based acoustic model at back-end. It is concluded that from table that ATP-LBP integrated with CQCC feature set outperforms all other feature set.

*5) **Performance of different Feature Integrations with DT-based Acoustic Model***

This section presents the results obtained using various proposed front end feature combinations with DT acoustic model at the back-end. As described earlier, three different DT model has been built by changing number of splits such as Fine, Medium, Coarse DT.

- **Performance of Integrated ATP and Audio Feature:** Table 14 gives the results for the proposed ASV system that uses ATP features integrated with audio features at front end and different types of DT based acoustic model at back-end. In is clearly observed from table that ATP-MFCC, ATP-PLP, ATP-CQCC, ATP-GTCC, and ATP-BFCC integrated feature produces 16, 40,34, 15, 27% EER using Fine DT, Medium DT, Fine DT, Fine DT, Fine DT, respectively. Hence, it is concluded that ATP-GTCC outperforms other feature set.

- **Performance of Integrated LBP and Audio Feature:** Table 15 gives the results for the proposed ASV system that uses LBP features integrated with audio features at front end and different types of DT based acoustic model at back-end. In is clearly observed that using Fine DT, Fine DT, Medium DT, Medium DT, Fine DT, the LBP-MFCC, LBP-PLP, LBP-CQCC, LBP-GTCC, LBP-BFCC integrated feature produces 25, 28, 32.5, 31, 24 per cent EER, accordingly. Hence, ATP-BFCC surpass other feature set.

- **Performance of Integrated ATP, LBP, and Audio Feature:** Table 16 gives the results for the proposed ASV system that uses ATP-LBP features integrated with audio features at front end and different types of NB based acoustic model at back-end. In is clearly observed that using Fine DT, Medium DT, Fine DT, Medium DT, ATP-LBP-MFCC, ATP-LBP-PLP, ATP-LBP-CQCC, ATP-LBP-GTCC, ATP-LBP-BFCC, integrated feature produces 16, 31.9, 36.4, 13, 34.5 per cent EER, accordingly. Hence, it is concluded that ATP-LBP-GTCC feature outperforms all other feature set.

*6) **Performance of different Feature Integrations with LSTM-based Acoustic Model***

This section presents the results obtained using various proposed front end feature combinations with LSTM based acoustic model at the back-end.

TABLE XXI. Comparison with Existing Methods

Work	Feature Extraction	Classifier	Replay Attack	Parameters				
				Accuracy %	Precision %	Recall %	F1-score %	EER
Chettri et al.[22]	CQCC IMFCC MFCC SDA SCMC	CNN, GMM Ensemble Model 1 Ensemble Model 2 Ensemble Model 3	No	NA	NA	NA	NA	2.64 9.57 9.57
Todisco et al. [5]	CQCC	Gaussian Mixture Model	Yes	NA	NA	NA	NA	2.2
Malik et al. [18]	GTCC ATP	SVM(ECOC)	Yes	99.1	99	99	99	1
Mittel et al. [26]	Static-Dynamic Hybrid CQCC	System 1: LSTM with time distributed wrapper System 2: LSTM 2DCNN	Yes	97.1	NA	NA	NA	2.9 0.9
Proposed Approach	ATP+LBP +MFCC ATP+LBP +PLP ATP+LBP +CQCC ATP+LBP +GTCC ATP+LBP +BFCC	LSTM	Yes	96 89 89 99.5 90	94 89 89 99 91	96 81 82 99.2 98	84 94 84 98 94	3 10 10 0.5 9

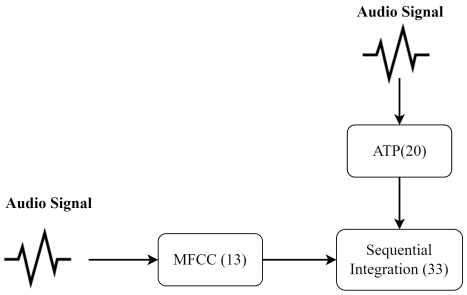
- **Performance of Integrated ATP and Audio Features:** Table 17 gives the results for the proposed ASV system that uses ATP features integrated with audio features at front end and different types of LSTM based acoustic model at back-end. In is clearly observed that ATP-MFCC, ATP-PLP, ATP-CQCC, ATP-GTCC, ATP-BFCC achieved EER 1.1%, 1.2%, 2.0%, 2.0%, 2.4%. Hence, it is concluded that ATP-MFCC outperforms other feature set.

- **Performance of Integrated LBP and Audio Features:** Table 18 gives the results for the proposed ASV system that uses LBP features integrated with audio features at front end and different types of LSTM based acoustic model at back-end. In is clearly observed that LBP-MFCC, LBP -PLP, LBP-CQCC, LBP-GTCC, LBP-BFCC achieved EER 3.3, 2.2, 1, 2, 3.9%. Hence, it is concluded that LBP-CQCC outperforms other feature set.

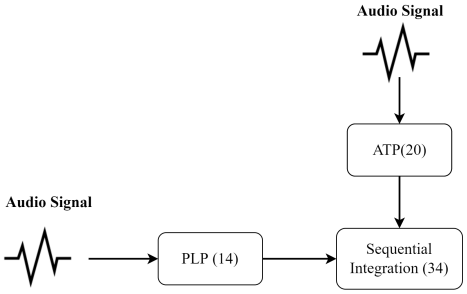
- **Performance of Integrated ATP, LBP, and Audio Features:** Table 19 gives the results for the proposed ASV system that uses ATP-LBP features integrated with audio features at front end and different types of LSTM based acoustic model at back-end. In is clearly observed that ATP-LBP-MFCC, ATP-LBP-PLP, ATP-LBP-CQCC, ATP-LBP-GTCC, ATP-LBP-BFCC achieved EER of 3, 10, 10, 0.5, 9%. Hence, it is concluded that ATP-LBP-GTCC outperforms other feature set.

5. Discussion and Comparative Analysis

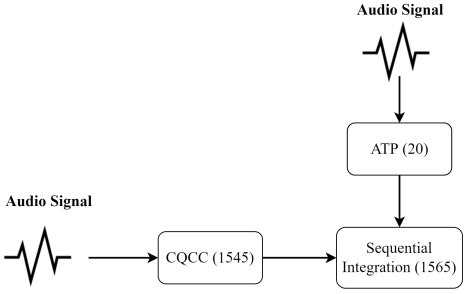
For enhancing the performance of the ASV systems, the hybrid feature extraction technique provides an improved way of extracting features from audio signals. The proposed work is an extension of some of the earlier proposed state-of-the-art works that have used hybrid features. Table 20 summarizes the results for the proposed ASV system that uses different feature combinations at front end and different types of acoustic models at back-end. It can be observed that ATP-LBP-GTCC over LSTM outperforms other feature sets. In our proposed work, five feature sets are created by combining ATP features with audio features. Similarly, five more feature sets are created by combining LBP image and audio features. These features are fed to LSTM and ML algorithms based acoustic models for classification. After



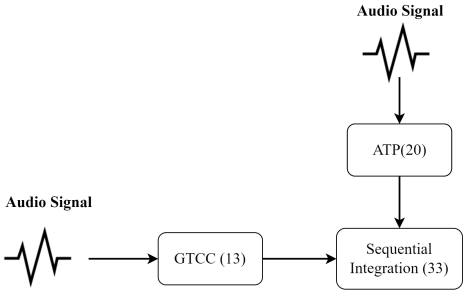
(a) ATP-MFCC Integration



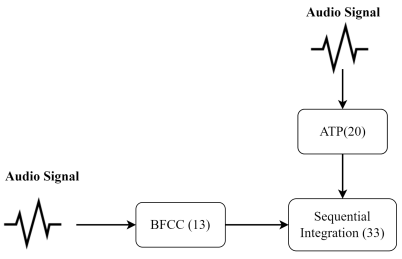
(b) ATP-PLP Integration



(c) ATP-CQCC Integration

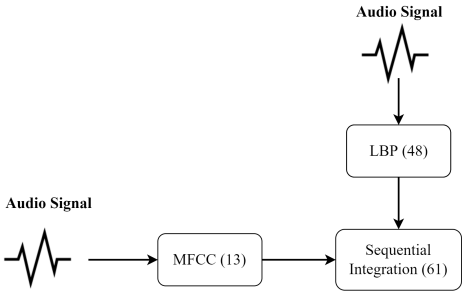


(d) ATP-GTCC Integration

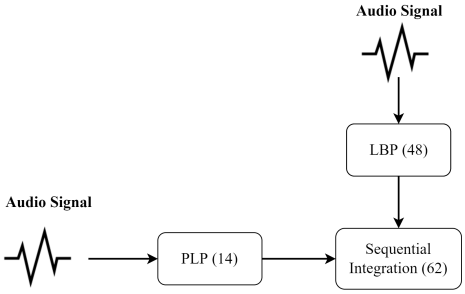


(e) ATP-BFCC Integration

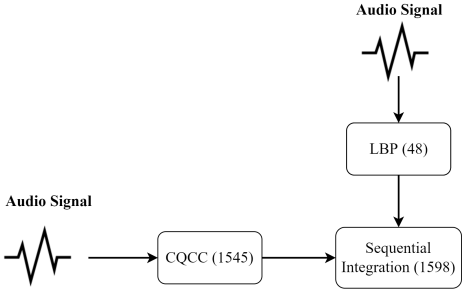
Figure 3. Integration of ATP with different Audio Feature Extraction Technique



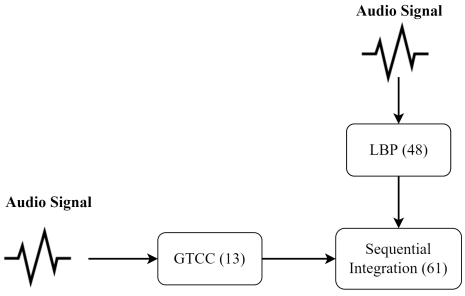
(a) LBP-MFCC Integration



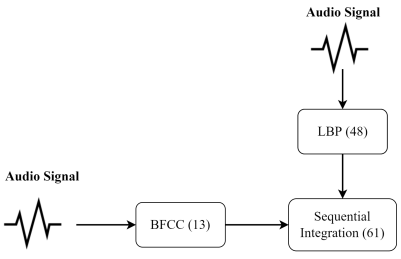
(b) LBP-PLP Integration



(c) LBP-CQCC Integration

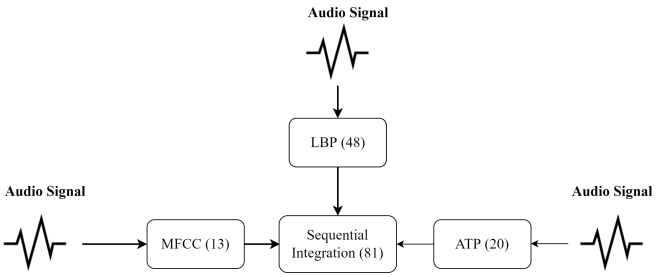


(d) LBP-GTCC Integration

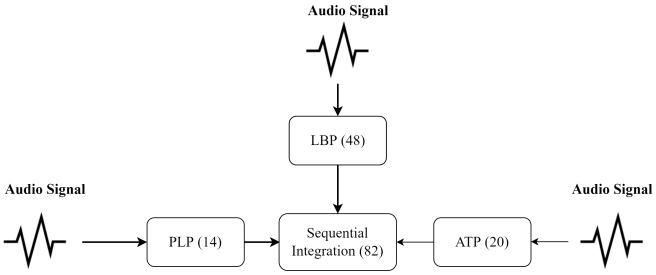


(e) LBP-BFCC Integration

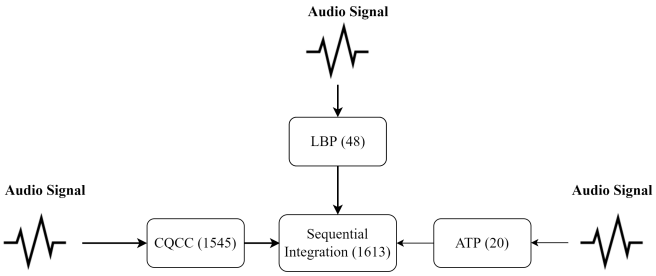
Figure 4. Integration of LBP with different Audio Feature Extraction Technique



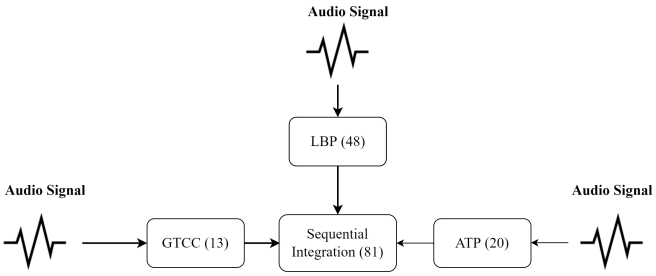
(a) ATP-LBP-MFCC Integration



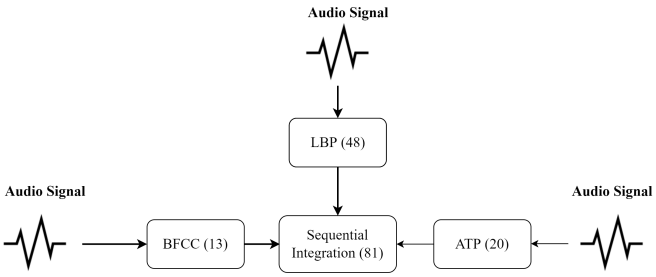
(b) ATP-LBP-PLP Integration



(c) ATP-LBP-CQCC Integration



(d) ATP-LBP-GTCC Integration



(e) ATP-LBP-BFCC Integration

Figure 5. Integration of ATP-LBP with different Audio Feature Extraction Technique

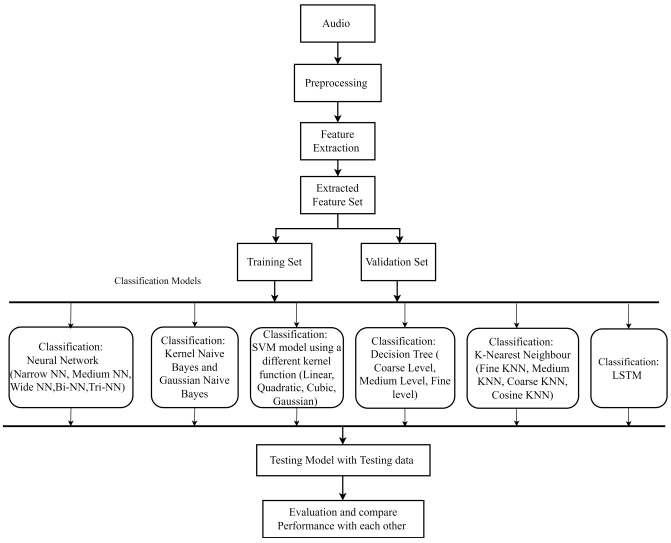


Figure 6. Process of training and evaluation of proposed model using classification models

comparing the result produced from these feature sets, it is observed that both ATP with GTCC and LBP with GTCC outperformed other models. Motivated by this, in the next experiment, both ATP-LBP features are combined with all audio features and again created five feature sets. From the result of the last investigation, it can be observed that ATP- LBP performed best with GTCC features. In the last decade, researchers have proposed various enhancements in front-end feature extraction methods and back-end acoustic models. As a result, significant improvements have been seen in various parameters used to measure the performance of ASV systems. Table 21 compares existing techniques with proposed work in terms of feature extraction method, back-end model and evaluation parameters used.

6. Conclusion and Future Work

In the contemplated work, The performance of the ASV system has been improved through the implementation of a hybrid FE method that integrates LSTM, thereby enhancing the system's ability to accurately recognize and verify the speaker's identity. Two image features, LBP and ATP, have been combined with various audio feature extraction techniques to form fifteen different feature combinations. Also, four different ML techniques such as NB, SVM, DT, and KNN have been used at the back end. Using the ASVspoof 2019 dataset, the suggested hybrid feature approach demonstrated excellent versatility and robustness. The feature set combination of ATP-LBP-GTCC with LSTM achieved the best performance with an EER of 0.5%. The proposed work can be extended by integrating some more image feature extraction techniques with already settled audio feature extraction techniques. More advanced attacks can be investigated, including mimicking, twin, deepfake, and spoofing attacks. Additionally, the use of augmentation techniques can be used to address the issue of uneven classification. Furthermore, to improve the performance of the proposed

ASV system, future work could involve the integration of diverse, cutting-edge datasets using advanced techniques, thereby potentially augmenting the system's accuracy and reliability.

References

[1] B. Beranek, "Voice biometrics: Success stories, success factors and what's next," *Biometric technology today*, vol. 2013, no. 7, pp. 9–11, 2013.

[2] A. De La Torre, J. C. Segura, C. Benitez, J. Ramirez, L. Garcia, and A. J. Rubio, "Speech recognition under noise conditions: Compensation methods," *Robust Speech Recognition and Understanding*, vol. 439, 2007.

[3] A. Mittal and M. Dua, "Automatic speaker verification system using three dimensional static and contextual variation-based features with two dimensional convolutional neural network," *International Journal of Swarm Intelligence*, vol. 6, no. 2, pp. 143–153, 2021.

[4] R. K. Aggarwal and M. Dave, "Performance evaluation of sequentially combined heterogeneous feature streams for hindi speech recognition system," *Telecommunication Systems*, vol. 52, no. 3, pp. 1457–1466, 2013.

[5] M. Todisco, X. Wang, V. Vestman, M. Sahidullah, H. Delgado, A. Nautsch, J. Yamagishi, N. Evans, T. Kinnunen, and K. A. Lee, "Asvspoof 2019: Future horizons in spoofed and fake audio detection," *arXiv preprint arXiv:1904.05441*, 2019.

[6] J. Yang, R. K. Das, and H. Li, "Extended constant-q cepstral coefficients for detection of spoofing attacks," in *2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. IEEE, 2018, pp. 1024–1029.

[7] Y. Qian, N. Chen, H. Dinkel, and Z. Wu, "Deep feature engineering for noise robust spoofing detection," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 10, pp. 1942–1955, 2017.

[8] G. K. Liu, "Evaluating gammatone frequency cepstral coefficients with neural networks for emotion recognition from speech," *arXiv preprint arXiv:1806.09010*, 2018.

[9] X. Valero and F. Alias, "Gammatone cepstral coefficients: Biologically inspired features for non-speech audio classification," *IEEE Transactions on Multimedia*, vol. 14, no. 6, pp. 1684–1689, 2012.

[10] T.-W. Kuan, A.-C. Tsai, P.-H. Sung, J.-F. Wang, and H.-S. Kuo, "A robust bfcc feature extraction for asr system." *Artif. Intell. Res.*, vol. 5, no. 2, pp. 14–23, 2016.

[11] M. Dua, R. K. Aggarwal, and M. Biswas, "Optimizing integrated features for hindi automatic speech recognition system," *Journal of Intelligent Systems*, vol. 29, no. 1, pp. 959–976, 2020.

[12] S. Joshi and M. Dua, "Lstm-gtcc based approach for audio spoof detection," in *2022 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COM-IT-CON)*, vol. 1. IEEE, 2022, pp. 656–661.

[13] O. K. Toffa and M. Mignotte, "Environmental sound classification using local binary pattern and audio features collaboration," *IEEE Transactions on Multimedia*, vol. 23, pp. 3978–3985, 2020.

[14] O. O. Khalifa, K. K. El-Darymli, A.-H. Abdullah, and J. I. Daoud, "Statistical modeling for speech recognition," 2013.

[15] F. Jelinek, "Continuous speech recognition by statistical methods," *Proceedings of the IEEE*, vol. 64, no. 4, pp. 532–556, 1976.

[16] X. Huang and M. Jack, "Performance comparison between semicontinuous and discrete hidden markov models of speech," *Electronics Letters*, vol. 24, no. 3, pp. 149–150, 1988.

[17] J. R. Bellegarda and D. Nahamoo, "Tied mixture continuous parameter modeling for speech recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 38, no. 12, pp. 2033–2045, 1990.

[18] T. Wang, Z. Liu, T. Zhang, S. F. Hussain, M. Waqas, and Y. Li, "Adaptive feature fusion for time series classification," *Knowledge-Based Systems*, vol. 243, p. 108459, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0950705122001903

[19] A. Kuamr, M. Dua, and T. Choudhary, "Continuous hindi speech recognition using gaussian mixture hmm," in *2014 IEEE Students' Conference on Electrical, Electronics and Computer Science*. IEEE, 2014, pp. 1–5.

[20] M. McLaren, R. Vogt, B. Baker, and S. Sridharan, "A comparison of session variability compensation techniques for svm-based speaker recognition," in *Proceedings of the 8th Annual Conference of the International Speech Communication Association (Interspeech 2007)*. Casual Productions Pty Ltd, 2007, pp. 790–793.

[21] K. M. Malik, A. Javed, H. Malik, and A. Irtaza, "A light-weight replay detection framework for voice controlled iot devices," *IEEE Journal of Selected Topics in Signal Processing*, vol. 14, no. 5, pp. 982–996, 2020.

[22] S. Escalera, O. Pujol, and P. Radeva, "Separability of ternary codes for sparse designs of error-correcting output codes," *Pattern Recognition Letters*, vol. 30, no. 3, pp. 285–297, 2009.

[23] M. Waqas, S. Tu, Z. Halim, S. U. Rehman, G. Abbas, and Z. H. Abbas, "The role of artificial intelligence and machine learning in wireless networks security: principle, practice and challenges," *Artificial Intelligence Review*, pp. 1–47, 2022.

[24] S. Tu, M. Waqas, Y. Meng, S. U. Rehman, I. Ahmad, A. Koubaa, Z. Halim, M. Hanif, C.-C. Chang, and C. Shi, "Mobile fog computing security: A user-oriented smart attack defense strategy based on dql," *Computer Communications*, vol. 160, pp. 790–798, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S014036642030253X

[25] Z. Wu and Z. Cao, "Improved mfcc-based feature for robust speaker identification," *Tsinghua Science & Technology*, vol. 10, no. 2, pp. 158–161, 2005.

[26] M. Witkowski, S. Kacprzak, P. Zelasko, K. Kowalczyk, and J. Galka, "Audio replay attack detection using high-frequency features." in *Interspeech*, 2017, pp. 27–31.

[27] B. Chettri, D. Stoller, V. Morfi, M. A. M. Ramírez, E. Benetos, and B. L. Sturm, "Ensemble models for spoofing detection in automatic speaker verification," *arXiv preprint arXiv:1904.04589*, 2019.

[28] M. Todisco, H. Delgado, and N. Evans, "Constant q cepstral coefficients: A spoofing countermeasure for automatic speaker verification," *Computer Speech & Language*, vol. 45, pp. 516–535, 2017.

[29] A. Mittal and M. Dua, "Constant q cepstral coefficients and long short-term memory model-based automatic speaker verification system," in *Proceedings of international conference on intelligent computing, information and control systems*. Springer, 2021, pp. 895–904.

[30] ——, "Static–dynamic features and hybrid deep learning models based spoof detection system for asv," *Complex & Intelligent Systems*, vol. 8, no. 2, pp. 1153–1166, 2022.

[31] H. Choudhary, D. Sadhya, and V. Patel, "Automatic speaker verification using gammatone frequency cepstral coefficients," in *2021 8th International Conference on Signal Processing and Integrated Networks (SPIN)*. IEEE, 2021, pp. 424–428.

[32] F. Noroozi, M. Marjanovic, A. Njegus, S. Escalera, and G. Anbarjafari, "Audio-visual emotion recognition in video clips," *IEEE Transactions on Affective Computing*, vol. 10, no. 1, pp. 60–75, 2017.

[33] K. Umapathy, S. Krishnan, and R. K. Rao, "Audio signal feature extraction and classification using local discriminant bases," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 4, pp. 1236–1246, 2007.

[34] P. Dhanalakshmi, S. Palanivel, and V. Ramalingam, "Classification of audio signals using svm and rbfnn," *Expert systems with applications*, vol. 36, no. 3, pp. 6069–6075, 2009.

[35] N. Chakravarty and M. Dua, "Noise robust asv spoof detection using integrated features and time delay neural network," *SN Computer Science*, vol. 4, no. 2, p. 127, 2022.

[36] T. Ojala, M. Pietikainen, and D. Harwood, "Performance evaluation of texture measures with classification based on kullback discrimination of distributions," in *Proceedings of 12th international conference on pattern recognition*, vol. 1. IEEE, 1994, pp. 582–585.

[37] N. Chatlani and J. J. Soraghan, "Local binary patterns for 1-d signal processing," in *2010 18th European Signal Processing Conference*. IEEE, 2010, pp. 95–99.

[38] A. Irtaza, S. M. Adnan, S. Aziz, A. Javed, M. O. Ullah, and M. T. Mahmood, "A framework for fall detection of elderly people by analyzing environmental sounds through acoustic local ternary patterns," in *2017 IEEE international conference on systems, man, and cybernetics (SMC)*. IEEE, 2017, pp. 1558–1563.

[39] S. M. Adnan, A. Irtaza, S. Aziz, M. O. Ullah, A. Javed, and M. T. Mahmood, "Fall detection through acoustic local ternary patterns," *Applied Acoustics*, vol. 140, pp. 296–300, 2018.

[40] X. Tan and B. Triggs, "Enhanced local texture feature sets for face recognition under difficult lighting conditions," *IEEE transactions on image processing*, vol. 19, no. 6, pp. 1635–1650, 2010.

[41] A. H. Abdelaziz, S. Zeiler, and D. Kolossa, "Learning dynamic stream weights for coupled-hmm-based audio-visual speech recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 5, pp. 863–876, 2015.

[42] D. Cooper, "Speech detection using gammatone features and one-class support vector machine," 2013.

[43] Y. Gong, J. Yang, and C. Poellabauer, "Detecting replay attacks using multi-channel audio: A neural network-based method," *IEEE Signal Processing Letters*, vol. 27, pp. 920–924, 2020.

[44] A. Godoy, F. Simoes, J. A. Stuchi, M. d. A. Angeloni, M. Uliani, and R. Violato, "Using deep learning for detecting spoofing attacks on speech signals," *arXiv preprint arXiv:1508.01746*, 2015.

[45] C. Hanilçi, T. Kinnunen, M. Sahidullah, and A. Sizov, "Classifiers for synthetic speech detection: A comparison," 2015.

[46] L. Lu, H.-J. Zhang, and S. Z. Li, "Content-based audio classification and segmentation by using support vector machines," *Multimedia systems*, vol. 8, no. 6, pp. 482–492, 2003.

[47] P. Wei, F. He, L. Li, and J. Li, "Research on sound classification based on svm," *Neural Computing and Applications*, vol. 32, no. 6, pp. 1593–1607, 2020.

[48] M. Z. Anwar, Z. Kaleem, and A. Jamalipour, "Machine learning inspired sound-based amateur drone detection for public safety applications," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 3, pp. 2526–2534, 2019.

[49] R. Thiruvengatanadhan, "Speech/music classification using mfcc and knn," *International Journal of Computational Intelligence Research*, vol. 13, no. 10, pp. 2449–2452, 2017.

[50] D. W. Aha, *A study of instance-based algorithms for supervised learning tasks: Mathematical, empirical, and psychological evaluations*. University of California, Irvine, 1990.

[51] M. Murugappan, "Human emotion classification using wavelet transform and knn," in *2011 International Conference on Pattern Analysis and Intelligence Robotics*, vol. 1. IEEE, 2011, pp. 148–153.

[52] S. K. Bhakre and A. Bang, "Emotion recognition on the basis of audio signal using naive bayes classifier," in *2016 International conference on advances in computing, communications and informatics (ICACCI)*. IEEE, 2016, pp. 2363–2367.

[53] A. Ghosh, N. Manwani, and P. Sastry, "On the robustness of decision tree learning under label noise," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 2017, pp. 685–697.

[54] Z. Kons, O. Toledo-Ronen, and M. Carmel, "Audio event classification using deep neural networks." in *Interspeech*, 2013, pp. 1482–1486.

Nidhi Chakravarty I, (corresponding author) completed her MTech in Computer Engineering from Centre for Development of Advanced Computing, NOIDA, 2020. She is pursuing Ph.D. in the area of Automatic Speaker Verification from National Institute of Technology, Kurukshetra.

Dr.Mohit Dua He received Ph.D. in the area of Automatic Speech Recognition from National Institute of Technology, Kurukshetra, India in 2018. He is presently working as Assistant Professor in Department of Computer Engineering at NIT Kurukshetra, India. He has more than 17 years of Teaching and Research experience. He is a member of Institute of Electrical and Electronics Engineers (IEEE), and life member of Computer Society of India (CSI) and Indian Society for Technical Education (ISTE). His research interests include Speech processing, Chaos based Cryptography, Information Security, Theory of Formal languages, Statistical modelling and Natural Language Processing. He has published approximately 60 research papers including abroad paper presentations including USA, Canada, Australia, Singapore, Mauritius and Dubai.