Automatic Speaker Verification using Machine Learning

Fakhare Alam
Department of Computer Science
Oakland University
Michigan, USA
fakherealam@oakland.edu

Tarun Gudela
Department of Computer Science
Oakland University
Michigan, USA
tgudela@oakland.edu

***Abstract*—Voice is projected to be the next input interface for portable devices. The increased use of audio interfaces can be mainly attributed to the success of speech and speaker recognition technologies. Automatic speaker verification (ASV) is one of the most natural and convenient means of biometric person recognition. Unfortunately, just like all other biometric systems, ASV is vulnerable to spoofing, also referred to as ”presentation attacks.” These vulnerabilities are generally unacceptable and call for spoofing countermeasures or ”presentation attack detection” systems. In addition to impersonation, ASV systems are vulnerable to replay, speech synthesis, and voice conversion attacks.This paper addresses these problems by building a speaker verification system using Machine Learning Algorithms to detect including spoofed data that cannot be differentiated from bona-fide utterances even by human subjects.**

***Index Terms*—ASV MFCC LPCC LLR, Voice Conversions, Text to speech, Replay Attacks, Amazon Alexa, Google Home**

I. Introduction

Automatic speaker verification (ASV) offers a low-cost and flexible biometric solution to person authentication.The reliability of ASV technology has advanced considerably recently and is currently deployed in a growing variety of practical applications such as Amazon Alexa, Google Home, Siri by Apple. Unfortunately, and as is the case for any biometric technology, concerns regarding vulnerabilities to spoofing, also referred to as presentation attacks, can undermine user confidence; thus, form a barrier to exploitation. By masquerading as another user, i.e. by mimicking their biometric traits,fraudsters can use spoofing attacks to infiltrate systems or services protected using biometric technology. Acknowledged spoofing attacks with regards to ASV include impersonation,replay, speech synthesis, and voice conversion as shown in **Fig 1(a)(b)**.

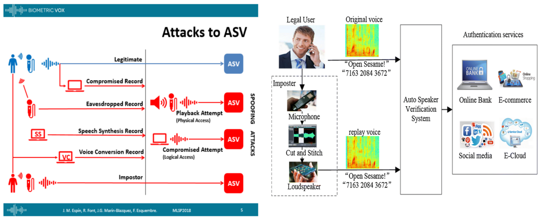


Fig. 1. Types of Attacks in ASV

In response to the threat of spoofing, researchers have sought to develop effective approaches to anti-spoofing and created multiple datasets for testing there Algorithms. We are working on one of the dataset which is typically generated using a limited number of specially crafted Spoofing - attack Algorithms along with main focus on Text to speech(TTS), Voice Conversions(VC) and the Replay spoofing attacks. This dataset is typically created for The 3rd Automatic Speaker Verification Spoofing and Countermeasures Challenge database (ASVSpoof 2019). This ASV Spoof 2019 Dataset aims to determine whether the advances in TTS and VC technology pose a greater threat to automatic speaker verification and the reliability of spoofing countermeasures. Advances with regards to the 2017 edition concern the use of a far more controlled evaluation setup for the assessment of replay spoofing countermeasures. Whereas the 2017 challenge was created from the recordings of real replayed spoofing attacks, the use of an uncontrolled setup made results somewhat difficult to analyse. A controlled setup, in the form of replay attacks simulated using a range of real replay devices and carefully controlled acoustic conditions, is adopted in ASVspoof 2019 with the aim of bringing new insights into the replay spoofing problem.

II. Literature Review

There are multiple researches going on in the field of Automatic speaker verification system almost from early 2009 and they have collected various forms of datasets to test their speaker verification systems over those datasets to test their working behaviour of their algorithms. with the collaboration of The Centre for Speech Technology Research(CSTR) at the University of Edinburgh they created a ASV spoof 2015 dataset which is created for the ASV spoofing and counter measures challenge. Genuine speech is collected from 106 speakers with no siginificant channel or background noises and the spoofed speech is generated by using the spoofing algorithms that are available at that time. This dataset helped many researches to test their algorithms. Like methods based on *Spectral Bitmap* was proposed to detect replay attacks and presented for text-dependent ASV system [2]. The other set of researches worked on finding out the comparison of the features for *Synthetic speech* detection using **GMM** and **SVM.** [3]. Along with this there is one more research which mainly concentrated on this dataset to evaluate the counter measures for speech synthesis and voice conversion spoofing attacks[4].

Using of Deep Neural Networks[5] to discriminate between different condition by identifying the changes in the playback,recording and environmental conditions by using *Constant-Q cepstral coefficients(CQCC)* and *High-Frequency cepstral coefficients(HFCC)*. In BTAS speaker anti-spoofing competition happened in 2016 there are many methods applied by many researches for detecting the spoofing in the audio files they are Spectogram-based ratios using logistic regression classifier, Two **GMMS** : *one for Genuine speech and the other for Spoofing attacks along with MFCC and inverted MFCC*[6]. Along with this there are so many approaches that are worked on the ASV spoof 2017 Dataset [7] which is mainly concentrated on Replay attack counter measures, use of GMM on the spectral centroid based frequency modulation features helped in detection of the replay attack along with some complementary features [8].

And the latest Dataset ASV spoof 2019 which we implemented is more advanced dataset which is generated using the state of the art neural acoustic and waveform models and the text to speech engines which are more realistic and very near to the human voice. And this dataset is divided into Logical access and Physical Access which was derived from the VCTK base corpus.

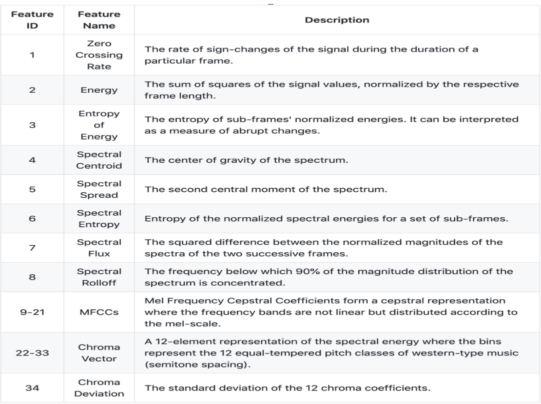
III. Dataset Description

ASVspoof 2019 adopts for the first time a new ASV-centric metric in the form of the tandem decision cost function (t-DCF). The ASVspoof 2019 database encompasses two partitions for the assessment of logical access (LA) and physical access (PA) scenarios. Both are derived from the VCTK base corpus which includes speech data captured from 107 speakers (46 males, 61 females). Both LA and PA databases are themselves partitioned into three datasets, namely training, development and evaluation which comprise the speech from 20 (8 male, 12 female), 10 (4 male, 6 female) and 48 (21 male, 27 female) speakers respectively. The three partitions are disjoint in terms of speakers, and the recording conditions for all source data are identical. While the training and development sets contain spoofing attacks generated with the same algorithms/conditions (designated as known attacks), the evaluation set also contains attacks generated with different algorithms/conditions (designated as unknown attacks). Reliable spoofing detection performance therefore calls for systems that generalise well to previously-unseen spoofing attacks.

IV. Feature Extraction

The audio files obtained form ASV Spoof 2019 dataset are compressed and are of .flac type. We followed the two step process to extract features from the audio files-

- **Step-1 Convert .flac to .wav files** - We used a wav converter tool available for Mac Operating System [10] to convert .flac files to wave files.
- **Step-2 Feature Extraction and Selection**-We tried tools such as JAudio [11], Matlab [12] and PyAudio[13]. After careful evaluations of features extracted from each tool and considering the features being used in current research, we used pyAudio to extract the feature. Finally, we selected the features mentioned in **Fig 2**.



Feature ID	Feature Name	Description
1	Zero Crossing Rate	The rate of sign-changes of the signal during the duration of a particular frame.
2	Energy	The sum of squares of the signal values, normalized by the respective frame length.
3	Entropy of Energy	The entropy of sub-frames' normalized energies. It can be interpreted as a measure of abrupt changes.
4	Spectral Centroid	The center of gravity of the spectrum.
5	Spectral Spread	The second central moment of the spectrum.
6	Spectral Entropy	Entropy of the normalized spectral energies for a set of sub-frames.
7	Spectral Flux	The squared difference between the normalized magnitudes of the spectra of the two successive frames.
8	Spectral Rolloff	The frequency below which 90% of the magnitude distribution of the spectrum is concentrated.
9-21	MFCCs	Mel Frequency Cepstral Coefficients form a cepstral representation where the frequency bands are not linear but distributed according to the mel-scale.
22-33	Chroma Vector	A 12-element representation of the spectral energy where the bins represent the 12 equal-tempered pitch classes of western-type music (semitone spacing).
34	Chroma Deviation	The standard deviation of the 12 chroma coefficients.

Fig. 2. Features and their description

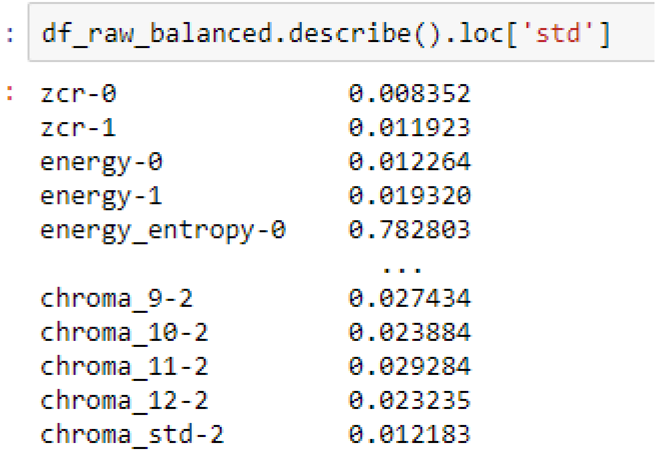
V. Methods

Once the feature data is extracted from the raw audio files, we followed life cycle of machine learning model building. At first, we performed Exploratory Data Analysis(EDA).Second, we used data prep-processing techniques to clean the data. In the third step, we build classification model using different machine learning algorithm. We used python[14] along with Jupyter Notebook IDE [15]for doing all the data pre-process steps ,EDA and model building. To increase the accuracy of the model, we iterated many times over model building process, which included modifying the feature subset and hyper tuning the model configuration to get the best performance out of individual classifier.

A. Exploratory Data Analysis (EDA)

The main aim EDA is to gain insights on the data, uncover missing values and discover underlying hidden patterns. After extracting the data, we performed following EDA -

- **Calculate Basic Statistics-** We checked the basic statistics of the dataset and found that there are total 443 features extracted initially but there is huge difference in the variance of each features. **Fig. 3** displays the variance in the feature set using standard deviation.
- **Check distribution of data w.r.t class label-** We plot the bar chart with respect to class label and found that dataset is highly imbalanced with respect to class label(Spoof, Bonafide). There are only 2580 audio files labelled as spoof. **Fig. 4** shows distribution of records against class lable.
- **Check for missing Values** -We found that there are lot of missing values w.r.t to features.**Fig. 5** shows the plot displaying the missing values. Yellow color corresponds to missing value with respect to each features.



```
: df_raw_balanced.describe().loc['std']

: zcr-0                0.008352
  zcr-1                0.011923
  energy-0             0.012264
  energy-1             0.019320
  energy_entropy-0     0.782803
                         ...
  chroma_9-2           0.027434
  chroma_10-2          0.023884
  chroma_11-2          0.029284
  chroma_12-2          0.023235
  chroma_std-2         0.012183
```

Fig. 3. Variance in feature subset

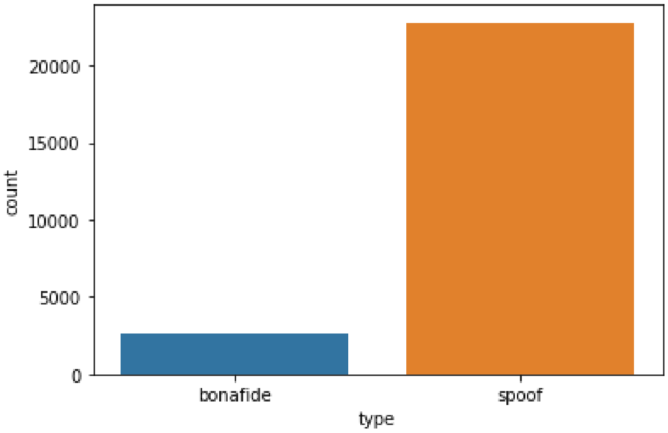


Fig. 4. Imbalance data set w.r.t class label (Bonafide,Spoof)

B. Data Pre-processing

Based on our exploratory analysis, we pre-processed data using follow steps-

- **Feature set Reduction and Impute Missing Values** - We removed all the features where more than 70% values are missing. In the next step, we imputed missing values for the remaining features with its average value. We performed this imputation because deleting all the features with missing values will also remove features containing important information. After this step the total feature set reduced to 102 features.
- **Balance Dataset-**One of the important finding obtained in EDA is that data is highly imbalance w.r.t class label.Training the model with imbalance dataset results in biased model towards spoofed detection. We needed a mechanism to balance the dataset either using under sampling or over sampling techniques. We explored oversampling technique(SMOTE)[16] by synthesizing the dataset to get more bonafide case. Synthesizing did not work well as it is very difficult to create bonafide audio records. After this, we used random under sampling technique, where cases from spoofed audio files are randomly taken to match equal number to bonafide case.

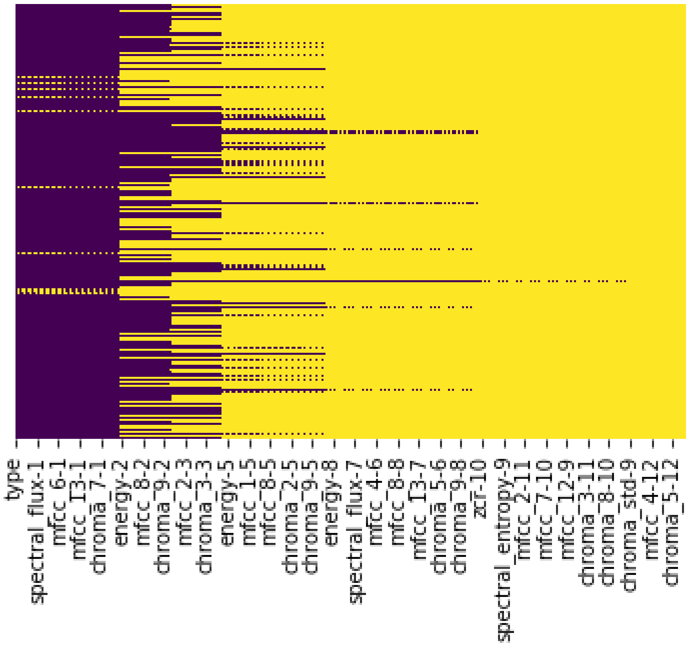


Fig. 5. Visualization -Missing Value in feature set

- **Test-Train Split**- The complete data was split into 80:20 ratios. For each classification method ,we used 80% of the data to train the model and remaining 20% to test the accuracy of the model. Later we performed 7-fold cross validation[17] to make sure that performance of the model is consistent across the dataset and good for future data.
- **Standardize the Dataset** In EDA, we found that there is huge difference in the feature variance. Basically features are on different scale. We utilized python in built standard scalar [18] to scale the values of each feature between 0 and 1
- **Principal Component Analysis (PCA)** After dropping feature with missing values, there are 104 features. We performed principal components analysis [19] on the scaled data to reduce the feature set by 30 and it explains more than 85% variance in the data. **Fig. 6** shows principal components and its explained variance.

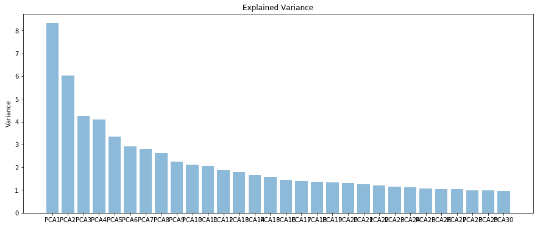


Fig. 6. Explained variance using PCA

C. Spoof Detection Model (SDM)

We used well known classification algorithm to build the model to detect spoof. We trained the model on 80% data, which was scaled and reduced to 30 principal components. First we created base model using logistic regression [20]. Later we built Spoof Detection Model using Support Vector Machine(SVM) [21], Adaptive Boosting(AdaBoost)[22], K-Nearest Neighbor (KNN)[23] and Artificial Neural Network (ANN)[24]. Each of this classifier are hyper-tuned to increase the accuracy on the test data. Model summary for each classifier is available on github[27]. In the last we built model using ensemble voting classifier [25] to accommodate the individual weakness of the model. We also performance 7-fold cross validation to ensure that model performance is consistent.

VI. Results

We measure individual model performance using four measures precision, recall and F1 Score [26]. **Table 1** displays the individual classifier performance along with ensemble model. The best preforming model is SVM using 'rbf' kernel with precision 95%, recall 95%, F1 Score 95% and accuracy 94.5.%. **Table 2** displays the model performance using 7-fold cross validation using mean accuracy and variance.

TABLE I
Machine Learning Model Performance

Classifier	Accuracy	Precision	Recall	F1 Score
Logistic Regression	0.89	0.90	0.90	0.90
AdaBoost	0.87	0.88	0.87	0.87
SVM*	0.945	0.95	0.95	0.95
KNN	0.92	0.92	0.92	0.92
ANN	0.928	0.93	0.93	0.93
Ensemble Model	0.937	0.94	0.94	0.94

TABLE II
7-Fold Cross validation Results

Classifier	Mean Accuracy	Variance
Logistic Regression	0.876	0.012
AdaBoost	0.863	0.009
SVM*	0.929	0.010
KNN	0.904	0.018
ANN	0.922	0.011
Ensemble Model	0.937	0.94

VII. Conclusion

In this paper, we propose a machine learning model to detect spoof using scaled and reduced audio frequency and time domain audio features. Audio spoof detection is important is today's time where generating spoof has become so easy using different software tool and cyber crime is on the rise in our increasingly digital world. More training data is needed to build a robust model and achieve more accuracy.

VIII. Challenges & Future Scope

We encountered following challenges in the implementation of spoof detection model-

- Conversion of all **.flac** files (25381) to **.wav** in one batch
- Exploring different software tools to get feature set.

In future we will improve the accuracy of spoof detection model by -

- Increase the feature subset by including Gamamtone features.
- Enlarge training set by providing audio files of sophisticated High quality spoof. This will make sure model is trained on diverse set of data.
- Identify the device configuration through which spoof was created and add that as feature in machine learning model.

References

[1] https://www.asvspoof.org/

[2] "A study on replay attack and anti-spoofing for text-dependent speaker verification." Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2014 Asia-Pacific. IEEE, 2014.

[3] Sahidullah, Md, Kinnunen, Tomi , Hanilçi, Cemal. (2015). A Comparison of Features for Synthetic Speech Detection.

[4] Wu, Zhizheng, et al. "ASVspoof: the automatic speaker verification spoofing and countermeasures challenge." IEEE Journal of Selected Topics in Signal Processing 11.4 (2017): 588-604.

[5] Nagarsheth, Parav, et al. "Replay Attack Detection Using DNN for Channel Discrimination." Interspeech. 2017.

[6] orshunov, Pavel, et al. "Overview of BTAS 2016 speaker anti-spoofing competition." 2016 IEEE 8th international conference on biometrics theory, applications and systems (BTAS). IEEE, 2016. 14.13-15 (2002): 1175-1220.

[7] Kinnunen, Tomi, et al. "The ASVspoof 2017 challenge: Assessing the limits of replay spoofing attack detection." (2017).

[8] Gunendradasan, Tharshini, et al. "Detection of Replay-Spoofing Attacks Using Frequency Modulation Features." Interspeech. 2018.

[9] Todisco, Massimiliano, et al. "Asvspoof 2019: Future horizons in spoofed and fake audio detection." arXiv preprint arXiv:1904.05441 (2019).

[10] https://apps.apple.com/us/app/to-wav-converter/id817915583?mt=12

[11] https://sourceforge.net/projects/jaudio/

[12] https://www.mathworks.com/

[13] https://pypi.org/project/PyAudio/

[14] https://www.python.org/

[15] https://https://jupyter.org/

[16] Chawla, N. V., Bowyer, K. W., Hall, L. O., Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. Journal of artificial intelligence research, 16, 321-357.

[17] https://scikit-learn.org/stable/modules/crossvalidation.html

[18] https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html

[19] https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html

[20] https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

[21] https://scikit-learn.org/stable/modules/svm.html

[22] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.AdaBoostClassifier.html

[23] https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html

[24] https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html

[25] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.VotingClassifier.html

[26] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.precision_recall_fscore_support.html

[27] https://github.com/fakharealam/CSI-6160/blob/master/CSI_6160_Project.ipynb