PROJECT REPORT
SYNTHETIC SPEECH ATTRIBUTION

Arpita Nema (202116004), Ambuj Mishra (202116003)

Dhirubhai Ambani-Institute of Information and Communication Technology, Gandhinagar, Gujarat

ABSTRACT

Recently, a lot of deep-fake audio techniques have been proposed in literature. With the advancement in this domain, it is also required to have an automatic algorithm for synthetic speech attribution. Such an algorithm is designed to determine which technique amongst a list of techniques has been used for speech synthesis. With this view, in this work, we propose a simple, end-to-end machine learning-based model which can identify which class of algorithm is used for synthetic audios creation. In the proposed algorithm, firstly, three important features are extracted, namely Mel-frequency cepstral coefficients(MFCC), Melspectogram and chroma_cqt. These features are used to train the support vector machine for classifying the audio clips. The proposed algorithm achieves accuracy of 97% on the clean audio clips. Also, the proposed algorithm takes less than one second to classify the audio clips.

***Index Terms*—** MFCC, Chroma_cqt, Melspectogram, Synthetic Audio, SVM.

1. INTRODUCTION

Today, a wide range of available approaches can be used to create fake synthetic speech audio tracks. Synthetic speech can be created using simple cut-and-paste waveform concatenation techniques. It can also be obtained using vocoders that use the source-filter model of the speech signal. Multiple methods for synthetic audio synthesis based on Convolutional Neural Networks (CNNs) have recently been proposed. These create incredibly realistic results that are difficult to distinguish from genuine speech, even when listening to it with human ears. In the literature, developing forensic detectors capable of differentiating real voice recordings from synthetically generated ones have received a lot of attention. Many published works like [1, 2, 3], are available to detect fake from the original. On the other hand, the issue of identifying a synthetic speech recording to the generator that created it, has received less attention. Knowing which algorithm was used to create a synthetic speech recording can be crucial in identifying the source of illegal content.

The goal of this project is to provide an audio recording of a synthetically generated speech track, determining which method was used to synthesize the speech from a list of candidates.

2. PROPOSED ALGORITHM

In this work, we make use of a machine learning method for classification. We got the inspiration to use SVM from work in [4] wherein they used SVM for audio event classification. However, in order to train any machine learning algorithm, we first need the important information from the data. Since, the processing on raw audio data is quite time-consuming and cumbersome, we instead represent the data with only important features. We extracted the audio features from the provided dataset. After concatenating these features, the SVM classification model is used to classify them into five classes. The architecture of the model is illustrated in Fig 1. It consists of the following steps:

2.1. Data Pre-processing

Dataset consists of 5000 audio files along with a .csv file containing the labels denoting the class to which each audio belongs.

2.2. Feature Extraction

Audio signal processing is a sub-part of signal processing, We need to extract features as it mainly processes the audio signals. By converting digital and analog signals, it reduces undesirable noise and balances the time-frequency ranges. We tried 6 different features individually and then took 4 different combination of top 3 best performing features. However, the best results we obtained were by concatenation of MFCC, Melspectogram and chroma_cqt features. We, then, have optimized parameters for best performing feature.

2.2.1. MFCC

Frequencies are not perceived on a linear scale by humans. Even though the gap is the same (i.e. '50 and 1,000 Hz' vs '10,000 and 10,500 Hz'), humans are better at recognizing variations in lower frequencies than in higher frequencies. Equal distances in pitch seemed equally distant to the

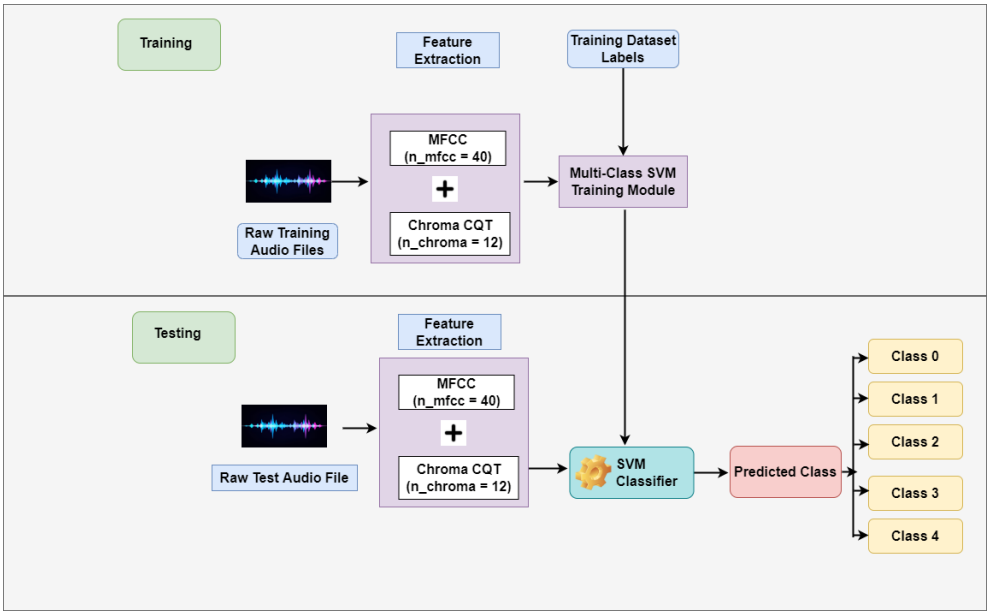

Fig. 1: Proposed Architecture.

listener on the Mel scale. Mel-Frequency Cepstral Coefficients (MFCCs) depict a sound's short-term power spectrum based on a Mel-scale transformation. It's often used in speech recognition because people's voices have a specific frequency range and differ from one another. In Librosa, retrieving and displaying MFCCs is a breeze.

2.3. Support Vector Machine (SVM) Classifier

In machine learning, SVM solves various regression and classification problems. The goal of the SVM method is to determine the best line or decision boundary for categorizing n-dimensional space in categories such that subsequent data points can be easily placed in the right category. SVM can be used for classification as well as pattern recognition applications of speech data and emotion data etc.

$$min_{w,b,D} \quad \frac{1}{2}W^T W + C\sum_{i=1}^{n} D_i \tag{1}$$

$$y_i(W^T\phi(x_i) + b) \geq 1 - D_i \tag{2}$$

where, C is regularization parameter,
D_i is the margin correction distance with $D_i \geq 0$, i=1...n,
$W^T W = ||W^2||$ denotes the normal vector,
$\phi(x_i)$ represents the transformed input vector space,
b represents the bias parameter,
y_i denotes the i-th target value.

For one-vs-one multi-class classification, the number of classifiers necessary can be retrieved with the following formula (with n being the number of classes and 5 for our problem):

$$n * (n-1)/2 \tag{3}$$

The objective is to find w and b such that most audios are predicted accurately. The kernel function explicitly maps every data point in the input space into a higher-dimensional space. In our solution, we used a linear kernel.

$$k(x_i, x_j) = x_i * x_j \tag{4}$$

In the proposed model, we made use of the simple SVM to cater to the given task. The performance of the SVM is the best on the dataset. Further, the model is also only simple, light-weighted, and easy to use. Initially, we divided the dataset into an 80/20 ratio to get the intuition about the performance. Also, we did a 10-fold stratified cross-validation of the whole data to get the generalized results.

Once we obtained our best-performing hyperparameters, feature vectors, and model, we re-trained our model on the entire dataset. This trained model was stored and later used to test the data given for validation. Similar to the earlier steps, the features from the test data were also computed and tested on the final saved model. The result was a multi-class label, which outputs the specific predicted class to which the given audio clip belongs.

3. RESULTS

To evaluate the performance of the model, we show the results in the following tables. We first evaluated the performance of the model on the basis of input features i.e., Mel-Frequency Cepstral Coefficient (MFCC), chroma_cqt (CH_Cqt), and Mel-Spectogram (MS) etc. We obtained the performance

on the evaluation dataset by taking different combinations of these features. In Table I, we show the accuracy scores achieved for clean, using different features applied on Support Vector Machine model for regularization parameter, C=1. As shown in the table, the highest accuracy is achieved in case of MFCC features i.e. 0.9606.

Table 1: The accuracy of SVM(C=1, kernel=linear) model on different audio features.

Features used and evaluation scores	
Features Used	***Accuracy***
MFCC	**0.9606**
Melspectogram	0.9204
chroma_stft	0.8398
chroma_stft_power_spectogram	0.8056
chroma_stft_energy_spectrum	0.8626
chroma_cqt	0.86

After selecting the best features which are individually performing better i.e., MFCC, Melspectogram and chroma_cqt, we measured the evaluation score after applying combination of these features to the complete dataset. We have used Support Vector Machine model for regularization parameter, C=1 and kernel=linear. SVM is more efficient than the other model in case of clean dataset. In Table-2, we can see that results for all three features concatenated are the best. Thus, we used SVM for our final proposed model.

Table 2: The accuracy of SVM(C=1, kernel=linear) model on different concatenated features.

Features used and evaluation scores	
Features Used	***Accuracy***
MFCC+chroma_cqt+Melspectogram	**0.9713**
chroma_cqt+Melspectogram	0.9349
MFCC+Melspectogram	0.9697
MFCC+chroma_cqt	0.9625

In order to show the effect of multiple regularization parameters (C) on the performance, in Fig. 2, we have taken different C parameter values, and measured the evaluation scores for that. As the plot depicts, our model works best for C=8 for linear kernel in case of all 3 concatenated features.

We performed a train-test split with 80% training data and 20% testing data on the complete dataset. We further tested our model on the testing dataset to create the confusion matrix. Fig. 3 shows the distribution amongst all classes on the testing dataset. From the confusion matrix, we see that for Classes 0, 3, and 4, the model works almost perfectly. However, most of the misclassification happens in Classes 1 and 2. In this view, when we heard the samples, we realized that

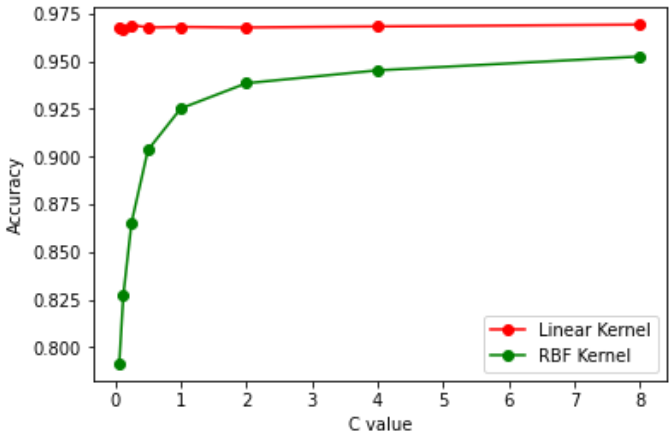

Fig. 2: Dependency of the proposed algorithm on different values of regularization parameter(C)

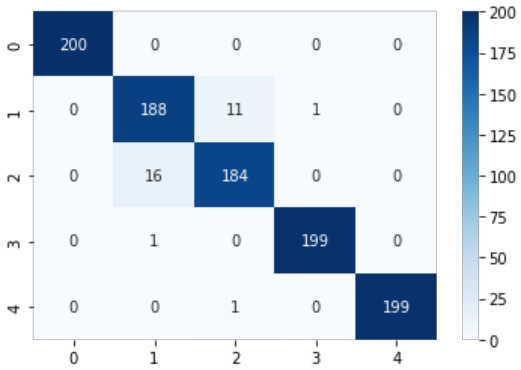

Fig. 3: Confusion matrix on 20% validation dataset

these two classes were perceptually very similar to each other. Hence, the same was depicted by the model.

Thus, our final proposed model consists of audio features MFCC, Melspectogram and chroma_cqt, trained on SVM model with C=8 and with kernel=linear.

Conclusion

In this work, we proposed a simple, light-weighted end-to-end machine learning-based model for classifying speech synthesis techniques. In this regard, we made use of some of the most important audio features and fed them to an SVM for training. We analyzed the basic SVM model for the given problem statement. Further, we also manipulated the various hyperparameters such as C , and found the model to give the optimal results on C equal to 8. We also found that for most of the classes, our model worked very well. However, for classes 1 and 2, there was a bit of misclassification. This was further confirmed when we analyzed these audio clips perceptually and found the clips very similar to each other.

4. REFERENCES

[1] R. Wijethunga, D. Matheesha, A. A. Noman, K. De Silva, M. Tissera, and L. Rupasinghe, "Deepfake audio detection: A deep learning based solution for group conversations," in *2020 2nd International Conference on Advancements in Computing (ICAC)*, vol. 1, 2020, pp. 192–197.

[2] T. Chen, A. Kumar, P. Nagarsheth, G. Sivaraman, and E. Khoury, "Generalization of audio deepfake detection," in *Proc. Odyssey 2020 The Speaker and Language Recognition Workshop*, 2020, pp. 132–137.

[3] A. Chintha, B. Thai, S. J. Sohrawardi, K. Bhatt, A. Hickerson, M. Wright, and R. Ptucha, "Recurrent convolutional structures for audio spoof and video deepfake detection," *IEEE Journal of Selected Topics in Signal Processing*, vol. 14, no. 5, pp. 1024–1037, 2020.

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