TECHNICAL REPORT
IEEE SIGNAL PROCESSING CUP 2022 (SP CUP)
SYNTHETIC SPEECH ATTRIBUTION

Arpita Nema, Ambuj Mishra, Fatima Naz, Pulkit Mahajan, Deebha Mumtaz, Vinit Jakhetiya

Indian Institute of Technology, Jammu, India

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, two important features are extracted, namely Mel-frequency cepstral coefficients(MFCC), and chroma_cqt. These features are used to train the support vector machine for classifying the audio clips. The proposed algorithm achieves validation accuracy of 92.54%, and 90.48% on the clean and noisy audio clips, respectively. Also, the proposed algorithm takes less than one second to classify the audio clips.

Index Terms— MFCC, Chroma_cqt, 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 the IEEE Signal Processing Cup 2022 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 main audio features from the provided dataset. After concatenating these features, the SVM classification model is used to classify them into six classes. The architecture of the model is illustrated in Fig 1. It consists of the following steps:

2.1. Data Pre-processing and Augmentation

Initially, we were provided with a clean dataset consisting of 6000 audio files along with a .csv file containing the information about the class to which each one belongs. Since the given dataset has no noise, we performed augmentation (compression, noise addition, and reverberation) on the given data. This helped to increase the data size as well as make the model more generalized and robust to degradation and noise. For noise addition, we have taken noise probability and SNR value as parameters, while bit-rate for compression. Figure 2, shows a flowchart for carrying out data augmentation. All the augmented files, along with the clean files were combined to form the final augmented dataset consisting of $24,000$ samples.

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.

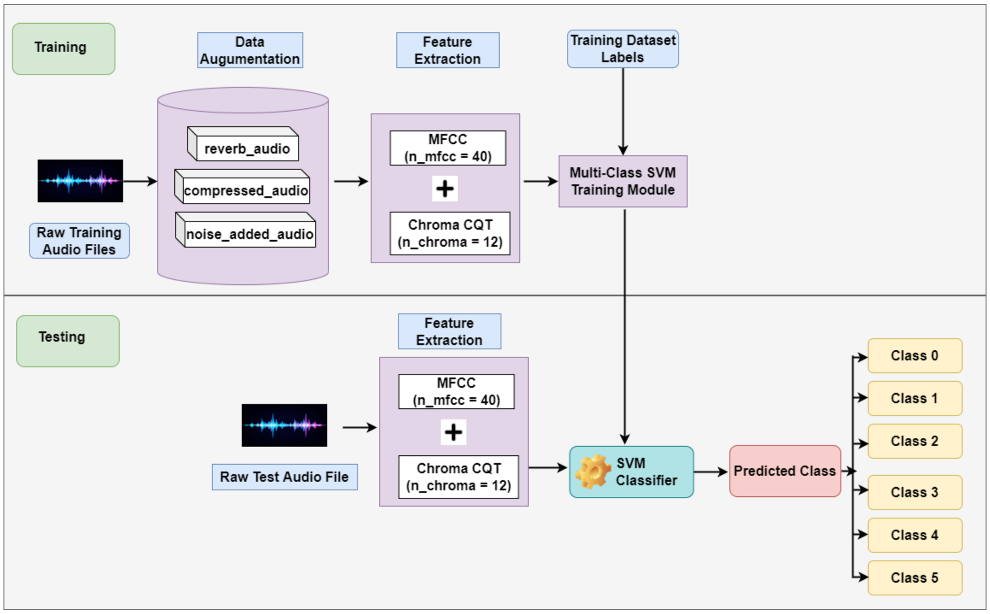

Fig. 1: Proposed Architecture.

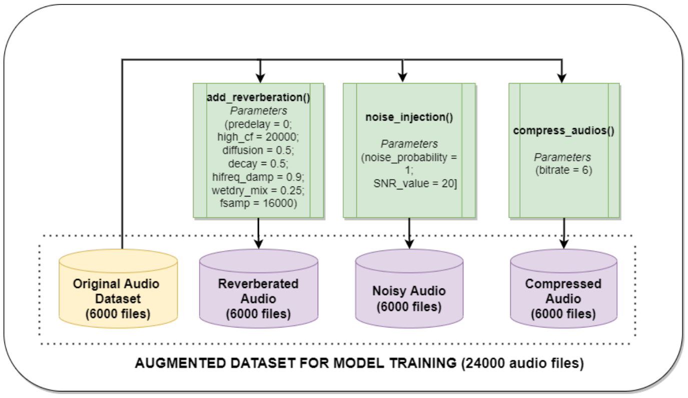

Fig. 2: Data Augumentation.

We tried many features individually and their combination like MFCCs, chroma_cqt, Mel-Spectrogram, spectral centroid, spectral contrast etc. However, the best results we obtained were by combining MFCC and chroma_cqt features.

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 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.

2.2.2. *Chroma_CQT*

Chroma feature visualization to know how dominant the characteristics of a certain pitch are present in the sampled frame. Chroma characteristics, which identify pitches that are different from one another by an octave, have high robustness as compared to tone fluctuations and are strongly related to the musical component of harmony. Because the terms chroma and pitch class are so closely related, chroma features are often named to as pitch class profiles.

We combined the MFCC and chroma_CQT features, each of dimension equal to 40 and 12 respectively, and fed this data into the next model for classification.

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.

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

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

where, C is regularization parameter,
D_i is the margin correction distance with $D_i \geq 0$, i=1...n,
W^TW = $||W^2||$ 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 6 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. Initially, though we applied a number of highly complex algorithms, the performance of the SVM was the best. 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 K-fold 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). We obtained the performance on the evaluation dataset by taking different combinations of these features. In Table I, we show the accuracy scores achieved from the Codalab, the scoring platform, for both clean and noisy data, using different combinations of features applied on Support Vector Machine model for regularization parameter, C=1. As shown in the table, the highest accuracy i.e. 0.9232, and 0.9027 is achieved using the combination of MFCC and chroma features. On the other hand, their individual performance is slightly lower than this.

After selecting the best features i.e., combination of MFCC and chroma, we measured the evaluation score after applying different algorithms to the complete augmented dataset. We have used four different types of models, which include; a simple 3-layer dense neural network (NN), support vector machine model (C=1), multistage hybrid model, and fusion model. In multi-stage hybrid model, the combination of SVM on stage-1 and Simple NN for binary classification of misclassified classes on stage-2 was carried on. In the fusion model, a combination of SVM and 3-Layer Simple NN is used to calculate the probability of each class for all audios in stage-1 and logistic regression to evaluate the final label in the final stage. As analyzed from the table, the performance of the results for the 3-layer dense neural network and SVM model on part-1 of the task isalmost equal, however, in the second task, the SVM is more efficient than the other model. Thus, we used SVM for our final proposed model.

Table 1: The accuracy of the proposed algorithm on different audio features.

Algorithm Name	**Features used and evaluation scores**		
	Features Used	***Codalab(p-1)***	***Codalab(p-2)***
SVM (C=1)	**MFCC+CH_Cqt**	**0.9232**	**0.9027**
SVM (C=1)	MFCC+CH_Cqt+MS	0.9175	0.8348
SVM (C=1)	MFCC	0.91857	-
SVM (C=1)	CH_Cqt	0.66	0.65

Table 2: Performance of different algorithms using the same concatenated features.

Algorithm Name	**Features used and evaluation scores**		
	Features Used	***Codalab(p-1)***	***Codalab(p-2)***
3-layer NN	MFCC+CH_Cqt+MS	0.9157	0.8761
SVM (C=1)	**MFCC+CH_Cqt+MS**	**0.9175**	**0.8348**
Hybrid model	MFCC+CH_Cqt+MS	0.9162	-
Fusion model	MFCC+CH_Cqt+MS	0.917	0.85

In order to show the effect of multiple regularization parameters (C) on the performance, in Fig. 4, we have taken different C parameter values, and measured the evaluation scores for that. As the plot depicts, our model works best for C=4.

We performed a train-test split with 80% training data and 20% validation data on the complete augmented dataset. We further tested our model on the validation dataset to create the confusion matrix. Fig. 5 shows the distribution amongst all classes on the validation dataset. From the confusion matrix, we see that for Classes 0, 3, and 4, the model works almost perfectly, while for class 5, there are just a few misclassified labels. However, most of the misclassification happens in Classes 1 and 2. In this view, when we heard the samples, we realized that these two classes were perceptually very similar to each other. Hence, the same was depicted by the model.

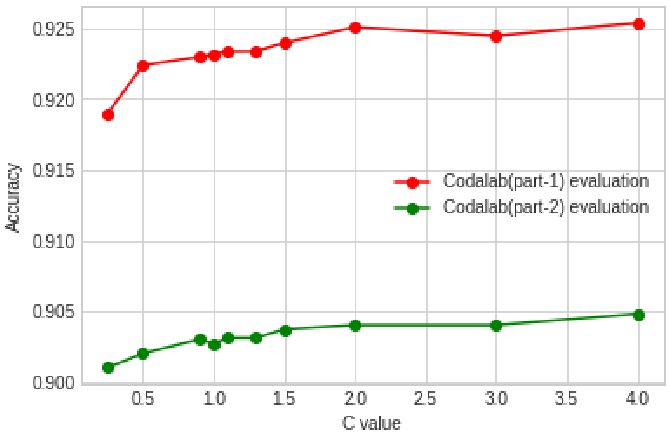

Fig. 3: Dependency of the proposed algorithm on different values of regularization parameter(C)

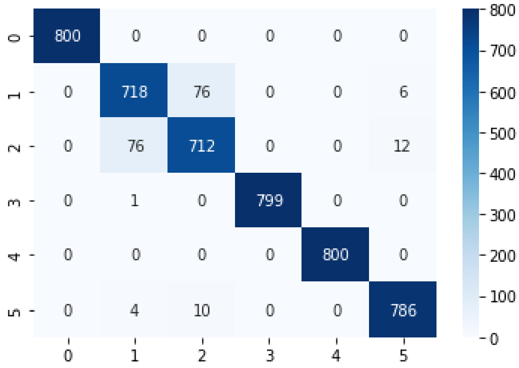

Fig. 4: Confusion matrix on 20% validation dataset

Thus, our final proposed model consists of audio features MFCC and chroma_cqt, trained on SVM model with C=4. The accuracy of the model is equal to 0.9254, and 0.9048 on part 1 and 2, respectively.

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 also experimented with a number of other ML techniques such as Random Forest, Dense layer architectures etc. However, on analysis we found the basic SVM model works the best 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 4. 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.