FGSSAT : Unsupervised Fine-Grain Attribution of Unknown Speech Synthesizers Using Transformer Networks

Kratika Bhagtani[†], Amit Kumar Singh Yadav[†], Ziyue Xiang[†], Paolo Bestagini[‡], and Edward J. Delp[†]

[†]Video and Image Processing Lab (VIPER), School of Electrical and Computer Engineering,
Purdue University, West Lafayette, Indiana, USA
[‡]Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Milano, Italy

***Abstract*—Synthetic speech generators can produce high quality speech. It can be difficult for humans to perceptually distinguish between synthesized speech and authentic human speech. Identifying the synthesizer used for generating synthetic speech, known as synthetic speech attribution, is an important problem. An open problem in synthetic speech attribution is attributing speech to new, *unknown* synthesizers, which are not present in the training set. Existing methods can identify *known* speech synthesizers but they cannot differentiate an *unknown* synthesizer from another *unknown* synthesizer. In this paper, we describe a system for attribution of *unknown* synthesizers *i.e.*, assigning different labels to different *unknown* synthesizers. Our system is known as Fine-Grain Synthetic Speech Attribution Transformer (FGSSAT). FGSSAT is unsupervised and uses transformer, dimensionality reduction and clustering for attribution. Our experiments use the ASVspoof2019 dataset. We train on real speech and 6 synthesizers and evaluate on real speech and 17 synthesizers, which include 11 *unknown* synthesizers. FGSSAT identifies *known* synthesizers with 99.6% accuracy and classifies all speech generated from *unknown* synthesizers with 76.5% accuracy, which is an improvement on existing work.**

***Index Terms*—Synthetic speech attribution, speech forensics, deepfake speech, transformer, unsupervised clustering**

I. Introduction

Recent deep learning methods can generate perceptually high quality and semantically consistent speech which can be used for commercials, Text-to-Speech (TTS) systems, movies, and voice assistants [1]–[3]. Synthetic speech generation tools can also be used for voice cloning or impersonating humans [4]. It is therefore possible to use these tools with malicious intent to commit financial fraud [5], and to spread misinformation [6]. The forensic community has focused on developing synthetic speech detection and attribution methods [7]–[9]. Synthetic speech detection methods can classify real speech from generated speech [7]. Synthetic speech attribution methods can identify the source of a speech signal [7].

Attributing speech signals from *known* synthesizers which are present during training, is known as closed set attribution. A major challenge in synthetic speech attribution is attributing speech generated by new, *unknown* synthesizers, for which prior information or speech for training is not available. This is known as open set attribution and is relevant since new speech generation methods are constantly developed.

Existing methods for synthetic speech attribution assign labels to speech generated from *known* speech synthesizers but they fail to differentiate between *unknown* speech synthesizers. They assign a single label to all speech signals generated from *unknown* synthesizers [10], [11]. We shall call this coarse-grain open set attribution. This limits attribution because we cannot distinguish between speech signals generated from different *unknown* synthesizers. Coarse-grain open set attribution also limits the answer to questions such as - was the same speech synthesizer used for multiple attacks targeting a person of interest? Has the synthesizer been seen before?

In this paper, we propose Fine-Grain Synthetic Speech Attribution Transformer (FGSSAT) to extend open set attribution to a fine-grain analysis for *unknown* speech synthesizers, *i.e.*, we assign different labels to speech signals generated from different *unknown* speech synthesizers. FGSSAT can also provide the number of different speech synthesizers (*known* and *unknown*). We evaluate our method on the ASVspoof2019 dataset [12]. Using FGSSAT, coarse-grain attribution is done on a per-speech signal basis, *i.e.*, the analyst can attribute one single speech signal to its source. Fine-grain attribution requires multiple speech signals to be present during testing. It can be useful in scenarios where an analyst analyzes attacks targeting a person of interest using multiple speech signals.

II. Related Work

Many synthetic speech detection methods have been proposed [13]–[16]. Some methods use transform coefficients for synthetic speech detection such as Constant-Q Cepstral Coefficients (CQCCs) [13] and Constant-Q Transform (CQT) [14]. Recent deep learning methods use either time-domain speech using a Recurrent Neural Network (RNN) [15] or the spectrogram representation of the speech using a neural network [16].

Synthetic speech attribution has been less investigated. Borrelli *et al*. [10] attribute speech signals to their synthesizers using Short-Term Long-Term (STLT) and Bicoherence features. Bartusiak *et al*. [11] and Yadav *et al*. [17] use transformer neural network for attribution. These methods are proposed for coarse-grain attribution, they classify all synthesizers not present in the training set as one *unknown* class.

Several transformer architectures have been proposed for image classification [18] and audio classification [19], [20].

Gong *et al*. proposed the Audio Spectrogram Transfomer [19] for audio classification tasks. Koutini *et al*. [20] proposed a transformer which uses patches of spectrogram for audio recognition. These transformers have shown high performance on synthetic speech detection [21], [22]. This motivated us to use Patchout faSt Spectrogram Transformer (PaSST) [20] for obtaining high-dimensional representations of the speech signal, which we further use for synthetic speech attribution as described in Section III.

III. Proposed Method

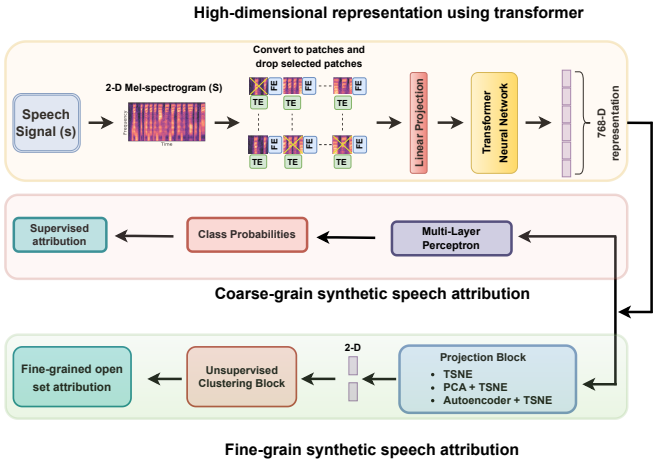

Fig. 1. Block diagram of our proposed method FGSSAT.

In this section, we describe our proposed method, FGSSAT. This section is divided into three parts. The first part describes obtaining high-dimensional representation of a speech signal. In the second part, we discuss how to use this representation for coarse-grain synthetic speech attribution. In the third part, we describe how to use this representation for fine-grain synthetic speech attribution.

High-dimensional representation: The block diagram of our proposed method is shown in Figure 1. We use time-domain speech signals with a fixed duration of 5 seconds. Similar to [17], [20], [22], we convert these input speech signals, denoted by **s** to a 2-D mel-spectrogram **S** [23] using the Short Time Fourier Transform [24] with Hanning window of duration 50ms and shift of 25ms. The mel-spectrogram **S** is a representation of the speech signal in 2D, where frequency is represented on the vertical axis on a mel-scale [23] and time is represented on the horizontal axis. If the frequencies are in the Hertz scale, it is known as a spectrogram [23].

The mel-spectrogram **S** is split into 2-D patches (Figure 1), and each patch is appended with a Time Encoding (TE) and a Frequency Encoding (FE) [20] as shown in Figure 1. Following [20], some of the patches are dropped during training, which reduces the computational complexity. Linear projection is applied to each patch, and together for all patches, a 768 dimensional representation of the input speech signal is obtained as output using a transformer neural network. The architecture of the transformer to obtain the 768-D representation is same as presented in [20]. It should be noted that the proposed transformer architecture was used for audio recognition in [20], here we are using the same type of transformer architecture for synthetic speech attribution. As shown in Figure 1, the transformer in FGSSAT is one part of the system and our contributions are in how the 768-D representation is further analyzed. The 768-D representation of the speech signal is used for both coarse-grain and fine-grain synthetic speech attribution (Figure 1) as described next.

Coarse-grain attribution: In coarse-grain synthetic speech attribution, our goal is to detect speech synthesizers present in the training set (*known* synthesizers) and to label all synthesizers not present in the training set as *unknown* synthesizers. In this case, all the speech synthesizers not present in the training set are labelled as one *unknown* class. We consider M *known* generators and every speech signal **s** has an associated synthesizer class label $y \in \{0, 1, 2, ..., M, M+1\}$. $y = 0$ represents real speech signals and $y = M + 1$ is the class label for all speech signals that are generated using *unknown* synthesizers. Following [10], during training the transformer network, two *known* synthesizers are labeled as *Known-Unknown* (KN-UNKN) [10] class *i.e.*, speech signals from these synthesizers are assigned $y = M + 1$. Although they are present in the training set, FGSSAT considers them as *unknown* (UNKN) synthesizers during training.

For every speech signal in coarse-grain attribution, its 768-D representation is passed as input to a Multi-Layer Perceptron (MLP) which is a modified version of the one used in [20] during training (see Section IV). The output of the MLP is a $(M+2)$-dimensional vector **P**. The i^{th} element of this vector is the probability that the input speech signal corresponds to the i^{th} class. Using the classification probabilities in **P**, the input speech signal is assigned the class label corresponding to the highest classification probability in **P**, *i.e.*, the estimated class label of the input speech signal $\hat{y} = \arg\max \mathbf{P}$.

Fine-grain attribution: Fine-grain synthetic speech attribution involves an unsupervised analysis. Our goal is to assign different labels to different *known* and *unknown* synthesizers. For fine-grain synthetic speech attribution, the 768-D representation obtained for each speech signal is used. This representation is passed through a projection block which projects it onto a 2-D Euclidean space. We use this projection block for dimensionality reduction because it was shown that processing and visualizing data representations requires less computation and memory in low-dimensional space [25]. We experimented with unsupervised clustering in multiple dimensions, and decided to project onto a 2-D Euclidean space, because of the better performance of the method in 2-D and better interpretability of the clusters. Hotelling described Principal Component Analysis (PCA), which is a dimensionality reduction method commonly used because it is linear, removes noise by reducing high-dimensional feature representations to low dimensions and produces uncorrelated features [26]. Bank *et al*. described that non-linear methods like Autoencoders often show better results for dimensionality

reduction than PCA [27]. Maaten *et al*. demonstrated the success of T-distributed Stochastic Neighbor Embedding (t-SNE) in reducing high-dimensional data to 2-D for perceptive visualization [28]. Because of its usefulness in dimensionality reduction to a 2-D space [28], we experiment with t-SNE and its combinations with other dimensionality reduction methods. We experiment with three approaches in the projection block:

- **T-distributed Stochastic Neighbor Embedding (t-SNE)** This non-linear dimensionality reduction method is described in detail in [28]. In our experiments for this approach, we use t-SNE [28] to reduce 768-D representations of the speech signals to 2-D representations.
- **Principal Component Analysis (PCA) and t-SNE** This low-dimensional representation using PCA is computed for all speech signals. We use PCA [26], [29], to reduce the 768-D representations of all speech signals to d-dimensional representations ($d = 50$ in our experiments because of high performance as compared to other values). Then, using t-SNE [28], these d-dimensional representations are further reduced to 2-D representations.
- **Autoencoder and t-SNE** In the third approach, instead of using PCA as we did in the second approach, we used a simple autoencoder [27], [30] for dimensionality reduction to d-dimensional representations of all speech signals. The input and output to the autoencoder are 768-D. The encoder and decoder consist of three fully-connected layers and are symmetrical. The bottleneck representation [27] that we obtain after the encoder is d-dimensional. We train the autoencoder to obtain the bottleneck representation such that the mean squared error between input and reconstructed output is minimum. After obtaining the d-dimensional bottleneck representation for all speech signals, we use t-SNE to further reduce the dimensions to 2-D.

The output of the projection block is a 2-D representation for each speech signal **s**. These are passed as input to the Unsupervised-Clustering Block which uses Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) method for generating clusters from the input [31]. Many other methods for clustering have been proposed. Centroid (cluster-center) based methods for example, k-means require specifying the required number of optimal clusters beforehand [32]. In the problem of synthetic speech attribution, we do not have prior knowledge of the number of speech synthesizers present. We use hierarchical density-based clustering because it does not require specifying the number of clusters, is not affected much by presence of noise in the data and its output is not dependent on the starting conditions [32]. HBDSCAN finds clusters by finding dense regions in the 2-D Euclidean space [31]. It uses a minimum spanning tree [31] and iteratively connects data points to their nearest neighbors [31]. Finally, the clustered output of the Unsupervised Clustering block provides the number of different speech synthesizers present (*known* and *unknown*).

TABLE I
ASVSPOOF2019 DATASET

Class Name	Train Set Signals	Dev Set Signals	Eval Set Signals	Method
BF	2580	2548	7355	
	Speech	**Generators**		
A01	3800	3716	-	NN
A02	3800	3716	-	VC
A03	3800	3716	-	VC
A04=A16	3800	3716 (A04)	4914 (A16)	WC
A05	3800	3716	-	VC
A06=A19	3800	3716 (A06)	4914 (A19)	VC
A07	-	-	4914	NN
A08	-	-	4914	NN
A09	-	-	4914	VC
A10	-	-	4914	NN
A11	-	-	4914	NN
A12	-	-	4914	NN
A13	-	-	4914	NN
A14	-	-	4914	VC
A15	-	-	4914	VC
A17	-	-	4914	VC
A18	-	-	4914	VC
Total	**25380**	**24844**	**71237**	

IV. EXPERIMENTAL RESULTS

A. The ASVspoof2019 Dataset

The distribution of the Logical Access (LA) part of the ASVspoof2019 Dataset [12] is described in Table I. This dataset consists of Bona fide (BF) speech signals and speech signals from 17 different synthesis methods. The entire dataset is divided into training, development and evaluation sets (referred to as Train, Dev and Eval, respectively). The speech synthesis methods in this dataset have been classified into three categories: Neural Network (NN), Vocoder (VC) and Waveform Concatenation (WC) based methods [12]. Synthesizer class A04 has the same generation method as A16, but is trained on a different training set. Since they both represent the same generation method, we consider them as one class in our experiments. The same argument holds for classes A06 and A19. For all our experiments, we train using the training set of the ASVspoof2019 dataset and test on a union of development and evaluation datasets. So, we have both *known* (A01 to A06) and *unknown* (A07 to A18 except A16 and A19) synthesizers in the testing set, which will be the case in a real-life scenario.

B. Experiments and Results

We used a pretrained transformer that was trained on the Audio Set Dataset [20], [33]. We then re-trained the transformer on the ASVspoof2019 dataset and obtained the 768-D representations as described above for all speech signals.

Experiment1 Coarse-grain Attribution: For this experiment, we use the Short-Term Long-Term (STLT) and Bicoherence feature method proposed by Borrelli *et al*. for comparison with FGSSAT [10]. In this feature-based method, the speech signal is divided into windows and the Bicoherence feature is determined using the Fourier transform of each

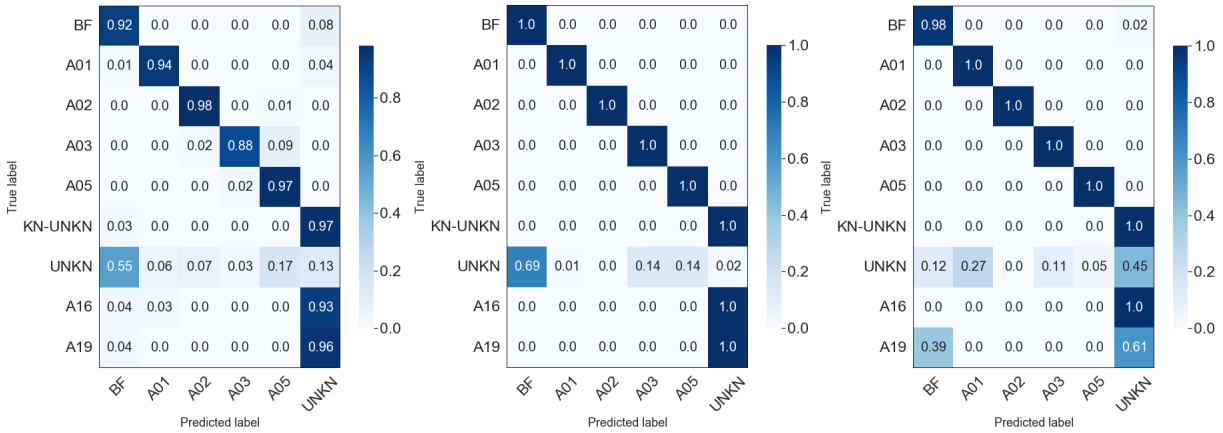

Fig. 2. Results of Experiment1 showing confusion matrices of the STLT and Bicoherence method (left) [10], SSAT (middle) [17] and FGSSAT (right).

window [10]. For determining the STLT features, Borrelli *et al*. model speech signals as auto-regressive processes [10]. We also compare our method with another transformer neural network method called Synthetic Speech Attribution Transformer (SSAT) [17]. In this method, the speech signals are converted to mel-spectrograms and a self-supervised pretrained transformer is used for attribution. For training the transformer network and MLP in FGSSAT, we follow a training approach similar to the STLT and Bicoherence feature method and the SSAT method. Two synthesizers from the training set (A04 and A06) are labeled KN-UNKN [10]. The remaining synthesizers in the training set (BF, A01, A02, A03 and A05) are *known*. They are classified as their respective class labels. All the other synthesizers in the testing set are *unknown*. Figure 2 shows the results of this experiment as confusion matrices. The results show that for all *known* classes, FGSSAT is at least 3 percentage points higher when compared to the STLT and Bicoherence feature method and has similar performance as compared to SSAT. A16 and A19 are classified as UNKN as they represent the same generation methods as A04 and A06, respectively which are also classified as UNKN. From Figure 2, we observe that FGSSAT correctly classifies 45% speech signals from the UNKN class (all speech signals not present in the training set), which is much higher as compared to the STLT and Bicoherence feature method and SSAT which correctly classify only 13% and 2% UNKN speech signals, respectively. Speech signals from KN-UNKN (speech signals present in the training set but treated as *unknown* during training), UNKN (all speech signals not present in the training set), A16, and A19 should be correctly classified as *unknown*. Using FGSSAT 76.75% of the *unknown* speech signals are classified as *unknown* as compared to the STLT and Bicoherence feature method and SSAT, in which 74.75% and 75.5%, respectively of the *unknown* speech signals are classified as *unknown*. Overall, for coarse-grain attribution, FGSSAT performs better than the STLT and Bicoherence feature method [10] and SSAT [17].

Experiment2 Fine-Grain Attribution: In this experiment, we use the 768-D representations of all speech signals in the testing set, experiment with all three dimensionality reduction approaches in the Projection Block, and cluster the obtained 2D representations using the Unsupervised Clustering Block. Note that to obtain the clustered output for fine-grain attribution, we do not use ground truth labels from the testing set.

We use three metrics for evaluating the performance of fine-

TABLE II
RESULTS OF EXPERIMENT2 SHOWING PERFORMANCE OF FGSSAT W.R.T THREE METRICS FOR UNSUPERVISED CLUSTERING FOR ALL THREE PROJECTION METHODS.

Projection Method	Silhouette Coefficient	Calinski Harabasz Index	Davies Bouldin Index
t-SNE	0.414	65509.665	0.645
PCA + t-SNE	0.352	73808.580	0.740
Autoencoder + t-SNE	0.267	51766.677	0.702

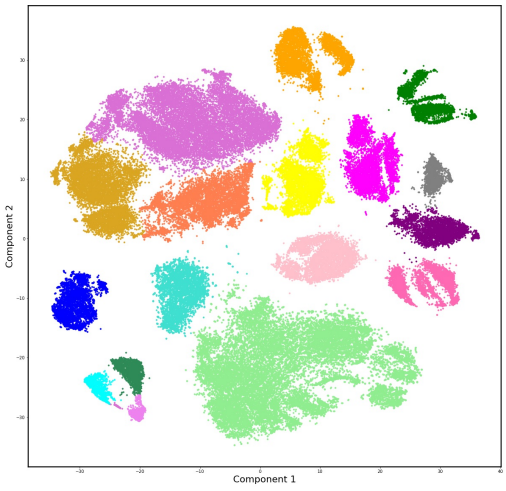

Fig. 3. Results of Experiment2 showing visualization of fine-grain attribution output using FGSSAT and t-SNE in the Projection Block. Each color represents a cluster. Component 1 and 2 represent the first and second element of the 2-D representation of each speech signal.

grain attribution with respect to characteristics of the clusters:

1. Silhouette Coefficient (SC) [34]: Higher SC implies well-separated, better-defined and dense clusters. For one speech signal, $SC = \frac{d_b - d_a}{\max(d_a, d_b)}$, d_a is the mean distance between the speech signal and other speech signals that belong to the same class (generated from same speech generator), d_b is the mean distance between the speech signal and other speech signals that belong to the next nearest cluster. In our experiments, we use the Euclidean distance measure for d_a and d_b. SC for the entire test set is the mean of SC's for all speech signals. SC can take values between -1 (wrong clustering) and 1 (correct and well-defined clustering).

2. Calinski-Harabasz Index (CHI) [35]: Higher CHI implies well-defined clusters. Let n_{total} be the total number of speech signals in the test dataset divided among k clusters, c_{total} be the center of the entire test dataset, n_i be the number of speech signals in the i^{th} cluster, C_i be the set of all speech signal representations in the i^{th} cluster and c_i be the center of i^{th} cluster. Let **x** represent the 2-D representation of a speech signal, we compute,

Within-cluster dispersion matrix $\mathbf{W_k}$:

$\mathbf{W_k} = \Sigma_{i=1}^{k} \Sigma_{\mathbf{x} \in \mathbf{C_i}} (\mathbf{x} - \mathbf{c_i})(\mathbf{x} - \mathbf{c_i})^T$

Between-cluster dispersion matrix $\mathbf{B_k}$:

$\mathbf{B_k} = \Sigma_{i=1}^{k} n_i (\mathbf{c}_i - \mathbf{c_{total}})(\mathbf{c}_i - \mathbf{c_{total}})^T$

CHI = $(\frac{tr(\mathbf{B_k})}{tr(\mathbf{W_k})}) \times (\frac{n_{total} - k}{k-1})$, $tr(\mathbf{B_k})$ represents trace of the matrix $\mathbf{B_k}$ and $tr(\mathbf{W_k})$ represents trace of the matrix $\mathbf{W_k}$.

3. Davies-Bouldin Index (DBI) [36]: Lower DBI implies well-separated clusters. Let the testing set is divided among k clusters and i and j represent indices of any two clusters ($i = 1, 2, ..k$ and $j = 1, 2, ..k$).

$R_{ij} = \frac{d_i + d_j}{d_{ij}}$ and DBI = $\frac{1}{k} \Sigma_{i=1}^{k} \max_{i \neq j} R_{ij}$

R_{ij} is a measure of similarity between clusters i and j, d_i and d_j are the mean distances between every point and center for clusters i and j, respectively, and d_{ij} is the distance between centers of clusters i and j.

We compute these three metrics (SC, CHI and DBI) for all three projection methods. Table II shows the performance of all three projection methods w.r.t these metrics. The results show that projecting 768-D representation of the speech signal into 2-D using t-SNE shows best performance in unsupervised clustering as compared to the other two projection methods. This is indicated by higher SC and lower DBI for t-SNE, as compared to the other two projection methods. The approach using PCA before t-SNE achieves next best results, followed by the approach using autoencoder before t-SNE.

We visualize the output of the Unsupervised clustering block for all three projection methods, and t-SNE gives the best visual results after clustering. Output of fine-grain attribution after clustering using t-SNE is shown in Figure 3. We observe that we obtain 15 clusters as output for 18 classes in the testing set as input. To evaluate the performance of FGSSAT for every class in the testing set, we use two other metrics as shown in Table III. To evaluate these metrics, we use the ground truth labels of all speech signals in the testing set. For a given class (*e.g.*, A08), we first find the cluster index which contains the majority of the speech signals belonging to that class (*e.g.*, A08). Using this cluster index we compute the two metrics:

TABLE III
RESULTS OF EXPERIMENT2 SHOWING CLASS-WISE PERFORMANCE OF FGSSAT FOR ALL THREE PROJECTION METHODS.

Class Label	% speech signals in one cluster	% speech signals from other synthesizers in that cluster
BF	**99.80%**	57.73%
A01	**99.62%**	**0.32%**
A02	46.31%	**0.12%**
A03	**100.00%**	**0.46%**
A04/A16	**85.34%**	**0.01%**
A05	**99.89%**	**0.24%**
A06/A19	**99.75%**	63.18%
A07	**68.62%**	76.50%
A08	**99.31%**	**1.71%**
A09	**68.11%**	**0.74%**
A10	**68.91%**	76.40%
A11	**99.59%**	**0.02%**
A12	**68.38%**	76.58%
A13	**69.43%**	76.22%
A14	**99.82%**	**0.65%**
A15	**98.27%**	**0.19%**
A17	**98.84%**	79.23%
A18	**99.24%**	**0.77%**

1. **% speech signals in one cluster:** We calculate the proportion of speech signals from the given class (*e.g.*, A08) which are assigned to the cluster index. For good performance, this metric should be as high as possible because all speech signals belonging to same class should be clustered together.

2. **% speech signals from other synthesizers in that cluster:** We calculate the proportion of speech signals from other classes (*e.g.*, all classes except A08) which are assigned to the cluster index. For good performance, this should be as low as possible because more than one class should not have speech signals clustered together.

These two metrics are calculated for the bona fide class and for each of the speech synthesizers present in the testing set. Table III shows the performance of FGSSAT w.r.t these two metrics for the first projection approach (t-SNE) and for all classes in the testing set. We observe that using the first projection approach *i.e.*, t-SNE, more than 65% speech signals that belong to one class are assigned to the same cluster for 17 out of 18 classes. For 11 classes, less than 2% of speech signals from other classes are assigned to the same cluster. Performance of t-SNE w.r.t the two metrics is higher as compared to the other two projection methods. FGSSAT can attribute and distinguish between 10 classes, which include both *known* and *unknown* speech synthesizers. Thus FGSSAT can be used for fine-grain attribution.

V. CONCLUSION

We proposed Fine-Grain Synthetic Speech Attribution Transformer (FGSSAT) that shows high performance for coarse-grain attribution. We also demonstrate its performance for fine-grain synthetic speech attribution, *i.e.*, ability to distinguish between different *unknown* speech synthesizers. In

future, we plan to explore neural network approaches for unsupervised analysis in fine-grain attribution.

ACKNOWLEDGEMENTS

This material is based on research sponsored by DARPA and Air Force Research Laboratory (AFRL) under agreement number FA8750-20-2-1004. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of DARPA and Air Force Research Laboratory (AFRL) or the U.S. Government. Address all correspondence to Edward J. Delp, ace@purdue.edu.

REFERENCES

[1] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "WaveNet: A Generative Model for Raw Audio," *Proceedings of the ISCA Workshop on Speech Synthesis Workshop*, p. 125, September 2016, Sunnyvale, USA.

[2] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan, R. A. Saurous, Y. Agiomvrgiannakis, and Y. Wu, "Natural TTS Synthesis by Conditioning Wavenet on MEL Spectrogram Predictions," *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 4779–4783, April 2018, Calgary, Canada.

[3] "Deep Learning for Siri's Voice: On-device Deep Mixture Density Networks for Hybrid Unit Selection Synthesis," August 2017. [Online]. Available: https://machinelearning.apple.com/research/siri-voices

[4] "Send in the clones: Using artificial intelligence to digitally replicate human voices," January 2022. [Online]. Available: https://www.npr.org/2022/01/17/1073031858/artificial-intelligence-voice-cloning

[5] T. Brewster, "Fraudsters Cloned Company Director's Voice In $35 Million Bank Heist, Police Find," October 2021. [Online]. Available: https://www.forbes.com/sites/thomasbrewster/2021/10/14/huge-bank-fraud-uses-deep-fake-voice-tech-to-steal-millions

[6] "Deepfake presidents used in Russia-Ukraine war," March 2022. [Online]. Available: https://www.bbc.com/news/technology-60780142

[7] K. Bhagtani, A. K. S. Yadav, E. R. Bartusiak, Z. Xiang, R. Shao, S. Baireddy, and E. J. Delp, "An Overview of Recent Work in Media Forensics: Methods and Threats," *arXiv:2204.12067*, April 2022.

[8] A. K. S. Yadav, K. Bhagtani, Z. Xiang, P. Bestagini, S. Tubaro, and E. J. Delp, "DSVAE: Interpretable Disentangled Representation for Synthetic Speech Detection," *arXiv:2304.03323*, July 2023.

[9] K. Bhagtani, E. R. Bartusiak, A. K. S. Yadav, P. Bestagini, and E. J. Delp, "Synthesized Speech Attribution Using The Patchout Spectrogram Attribution Transformer," *Proceedings of the ACM Workshop on Information Hiding and Multimedia Security*, pp. 157–162, June 2023.

[10] C. Borrelli, P. Bestagini, F. Antonacci, A. Sarti, and S. Tubaro, "Synthetic speech detection through short-term and long-term prediction traces," *EURASIP Journal on Information Security*, vol. 2021, no. 1, April 2021.

[11] E. R. Bartusiak and E. J. Delp, "Transformer-Based Speech Synthesizer Attribution in an Open Set Scenario," *Proceedings of the IEEE International Conference on Machine Learning and Applications*, December 2022, Nassau, The Bahamas.

[12] J. Yamagishi, M. Todisco, M. Sahidullah, H. Delgado, X. Wang, N. Evans, T. Kinnunen, K. A. Lee, V. Vestman, and A. Nautsch, "ASVspoof 2019: The 3rd Automatic Speaker Verification Spoofing and Countermeasures Challenge database," *University of Edinburgh. The Centre for Speech Technology Research*, March 2019.

[13] 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, September 2017.

[14] X. Li, N. Li, C. Weng, X. Liu, D. Su, D. Yu, and H. Meng, "Replay and Synthetic Speech Detection with Res2Net Architecture," *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 6354–6358, June 2021, Toronto, Canada.

[15] G. Hua, A. B. J. Teoh, and H. Zhang, "Towards End-to-End Synthetic Speech Detection," *IEEE Signal Processing Letters*, vol. 28, pp. 1265–1269, June 2021.

[16] E. R. Bartusiak and E. J. Delp, "Synthesized Speech Detection Using Convolutional Transformer-Based Spectrogram Analysis," *Proceedings of the IEEE Asilomar Conference on Signals, Systems, and Computers*, pp. 1426–1430, October 2021, Asilomar, CA.

[17] A. K. S. Yadav, E. Bartusiak, K. Bhagtani, and E. J. Delp, "Synthetic speech attribution using self supervised audio spectrogram transformer," *Proceedings of the IS&T Media Watermarking, Security, and Forensics Conference, Electronic Imaging Symposium*, January 2023.

[18] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," *arXiv:2010.11929*, June 2021.

[19] Y. Gong, Y.-A. Chung, and J. Glass, "AST: Audio Spectrogram Transformer," *Proceedings of the ISCA Interspeech*, pp. 571–575, August 2021, Brno, Czech Republic.

[20] K. Koutini, J. Schlüter, H. Eghbal-zadeh, and G. Widmer, "Efficient Training of Audio Transformers with Patchout," *Proceedings of Interspeech*, pp. 2753–2757, September 2022, Incheon, Korea.

[21] E. R. Bartusiak, "Machine Learning for Speech Forensics and Hypersonic Vehicle Applications," Ph.D. dissertation, Purdue University, West Lafayette, IN, 12 2022.

[22] A. K. Singh Yadav, Z. Xiang, E. R. Bartusiak, P. Bestagini, S. Tubaro, and E. J. Delp, "ASSD: Synthetic Speech Detection in the AAC Compressed Domain," *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 1–5, June 2023, Greece.

[23] S. S. Stevens, J. Volkmann, and E. B. Newman, "A Scale for the Measurement of the Psychological Magnitude Pitch," *Journal of the Acoustical Society of America*, vol. 8, no. 3, pp. 185–190, June 1937.

[24] L. R. Rabiner and R. W. Schafer, *Theory and Applications of Digital Speech Processing*, 1st ed. USA: Prentice Hall Press, 2010.

[25] W. Jia, M. Sun, J. Lian, and S. Hou, "Feature dimensionality reduction: a review," *Complex & Intelligent Systems*, vol. 8, no. 3, pp. 2663–2693, January 2022.

[26] H. Hotelling, "Analysis of a complex of statistical variables into principal components," *Journal of Educational Psychology*, vol. 24, no. 6, p. 417–441, 1933.

[27] D. Bank, N. Koenigstein, and R. Giryes, "Autoencoders," *arXiv:2003.05991*, April 2021.

[28] L. van der Maaten and G. Hinton, "Visualizing High-Dimensional Data Using t-SNE," *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, November 2008.

[29] K. P. F.R.S., "On lines and planes of closest fit to systems of points in space," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 2, no. 11, pp. 559–572, 1901.

[30] U. Michelucci, "An Introduction to Autoencoders," *arXiv:2201.03898*, January 2022.

[31] D. M. Ricardo J. G. B. Campello and J. Sander, "Density-Based Clustering Based on Hierarchical Density Estimates," *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, vol. 7819, pp. 160–172, April 2013.

[32] M. Omran, A. Engelbrecht, and A. Salman, "An overview of clustering methods," *Intelligent Data Analysis*, vol. 11, pp. 583–605, November 2007.

[33] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio Set: An ontology and human-labeled dataset for audio events," *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 776–780, June 2017.

[34] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *Journal of Computational and Applied Mathematics*, vol. 20, pp. 53–65, November 1987.

[35] T. Caliński and J. Harabasz, "A dendrite method for cluster analysis," *Communications in Statistics*, vol. 3, no. 1, pp. 1–27, January 1974.

[36] D. L. Davies and D. W. Bouldin, "A Cluster Separation Measure," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-1, no. 2, pp. 224–227, April 1979.