U.P.B. Sci. Bull., Series C, Vol. 88, Iss. 1, 2026 ISSN 2286 – 3540

DETECTING SYNTHETIC SPEECH WITH CONVOLUTIONAL NEURAL NETWORKS AND ADVANCED TRAINING METHODS

Irina MUTICA[1], Serban MIHALACHE[2], Dragos BURILEANU[3]

The rapid progress of text-to-speech (TTS) and voice conversion (VC) technologies has enabled the creation of synthetic voices that closely mimic real speakers. While these systems have beneficial applications, they also raise security concerns, being susceptible to misuse by fraud, impersonation, and disinformation. In this paper, we propose a synthetic speech detection system based on conventional convolutional neural network (CNN) models trained from scratch on cepstral-based features. We conduct separate experiments using Mel-frequency, linear frequency, and constant-Q cepstral coefficients (MFCCs, LFCCs, CQCCs) as inputs. To improve robustness and reduce overfitting, we employ targeted augmentation methods (Mixup, SpecAugment) alongside regularization techniques including batch normalization, dropout, label smoothing, and adaptive learning rate scheduling. Our experiments are carried out on the Fake-or-Real (FoR) corpus, a large-scale benchmark that includes both authentic human recordings and outputs from a variety of TTS and VC systems. Our best system configuration achieves 99.4% validation accuracy and 95.9% test accuracy, outperforming our previous work and demonstrating that carefully designed CNN architectures combined with focused augmentation and regularization can deliver high detection performance.

Keywords: deep neural networks; deepfake audio; convolutional neural networks; FoR dataset; synthetic speech detection

1. Introduction and Related Work

Over the past decade, speech synthesis has shifted from rule-based and statistical approaches to deep learning methods capable of producing natural, human-like voices. Modern TTS engines and neural voice cloning frameworks can generate speech that is difficult to distinguish from authentic recordings, raising challenges for security and media integrity. These frameworks allow synthesizing

[1] Ph.D. student, Speech and Dialogue Research Laboratory (SpeeD), National University of Science and Technology POLITEHNICA Bucharest, Romania, e-mail: irina.mutica@stud.etti.upb.ro

[2] Lecturer, Speech and Dialogue Research Laboratory (SpeeD), National University of Science and Technology POLITEHNICA Bucharest, Research Institute for Artificial Intelligence “Mihai Draganescu”, Romanian Academy, Romania, e-mail: serban.mihalache@upb.ro

[3] Professor, Speech and Dialogue Research Laboratory (SpeeD), National University of Science and Technology POLITEHNICA Bucharest, Romania, e-mail: dragos.burileanu@upb.ro

human-quality speech from text or cloning a speaker's voice with only seconds-long samples, making it ever harder to tell genuine from fake audio [1-2].

While deepfake video detection has been widely studied, synthetic speech forensics remains relatively underexplored. Early detectors relied on hand-crafted features, e.g., MFCCs, and simple classifiers [3-5], whereas recent efforts have gravitated toward end-to-end CNNs with domain adaptation, such as light CNNs with a "genuinization" transformer trained only on genuine speech before spoof detection [6], siamese CNNs that model Gaussian-probability features for fine-grained spoofing detection [7], and long-term high frequency descriptors such as inverted CQT-based cepstral and block coefficients, which are able to capture artifacts beyond the reach of short-term windows [8].

Furthermore, malicious actors can splice brief synthetic segments into genuine recordings, subtly altering key phrases to evade standard detectors and distort messages in fraud, social engineering, or disinformation campaigns. In [9], this is tackled by extracting frame-level CQCC features and feeding them into an artificial neural network-based detector, followed by Intersection-over-Union (IoU) driven post-processing to localize the tampered segments. This work underscores the power of CQCCs for capturing fine-grained spoofing artifacts.

Robustness in open-set scenarios has been advanced through adaptive reliability estimation, selecting analysis windows based on per-window confidence [10], and by modeling local autoregressive coefficients with variance statistics [11]. Foundational efforts include one-class learning frameworks for enhanced generalization [12], dual-branch architectures for logic-manipulated speech detection [13], transformer-based classifiers resilient to compression artifacts [14], and ensemble schemes that fuse GMM-ResNet2 features for higher accuracy [15]. More recent work has investigated specialized backbones and attention mechanisms, such as integrating spatially reconstructed local attention into Res2Net to exploit discriminative subband cues [16] and spectrogram-based analyses comparing constant-Q and IIR filter transforms to capture complementary time-frequency information [17].

One ongoing challenge is robustness over time: as TTS and VC algorithms evolve, detection models must adapt to new voices, codecs, and real-world noise conditions. Moreover, the limited variety of high-fidelity spoof datasets constrains a model's exposure during training, hindering cross-domain generalization.

In our previous work [1-2], we tested 3 deep learning systems on the FoR dataset: an MLP leveraging hand-crafted spectral features; a CNN on the same hand-crafted spectral features; and an EfficientNetV2 backbone fine-tuned via transfer learning on raw spectrograms. Without any data augmentation or modern regularization, our previous CNN-based model reached 98.9% validation and 83.9% testing accuracy [1], while our augmented transfer learning-based approach achieved higher performance, with 97.5% validation and 91.1% test accuracy [2].

Building on these insights, the present study forgoes transfer learning and instead improves upon the CNN architecture while adding several regularization and augmentation techniques, trained directly on MFCC, LFCC, and CQCC feature vectors as inputs. We integrate SpecAugment time/frequency masking, Mixup, and advanced regularization (dropout, weight decay, label smoothing, batch normalization). To further refine training, we explore configurable cosine decay and one-cycle learning rate schedules.

The main contributions of this work include:

- a CNN-based pipeline trained directly on MFCC, LFCC, and CQCC inputs, strengthened with targeted data augmentation and advanced regularization techniques;
- an extensive evaluation on the Fake-or-Real (FoR) dataset, covering genuine speech and synthetic speech samples generated by recent TTS/VC models, showing improved accuracy and robustness.

The rest of the paper is organized as follows: section 2 describes the proposed system architecture; section 3 provides methodological details for the experimental setup, as well as the obtained results and their discussion; conclusions and future work are outlined in section 4.

2. Proposed Methodology

The architecture of the proposed synthetic speech detection system, illustrated in Fig. 1, consists of a conventional convolutional neural network (CNN) receiving as input a feature vector derived from algorithmically extracted cepstral coefficients. In this work, only one type of cepstral features is used per experiment: Mel-frequency cepstral coefficients (MFCCs), linear frequency cepstral coefficients (LFCCs), or constant-Q cepstral coefficients (CQCCs). For each utterance, the selected coefficients are computed for all frames and concatenated with their deltas and delta-delta coefficients, after which statistical functions (mean and standard deviation) are applied across time to form the final input vector.

Fig. 2 shows frame-level heatmaps of MFCC, LFCC, and CQCC features for two utterances, illustrating the distinct spectral emphasis each coefficient type provides. MFCCs highlight the spectral envelope, with lower indices tracking formant dynamics and higher indices encoding finer spectral detail. LFCCs, using a linear-scale filterbank, distributes resolution more evenly across the spectrum, revealing high frequency detail more uniformly [18]. CQCCs, computed from a constant-Q transform, capture both envelope cues and harmonic micro-structure, often visible as fine, quasi-periodic textures. Visual differences, such as more repetitive mid/high-order patterns in the synthetic sample versus richer temporal variability in the real sample, motivate combining these representations in the proposed system [19-20].

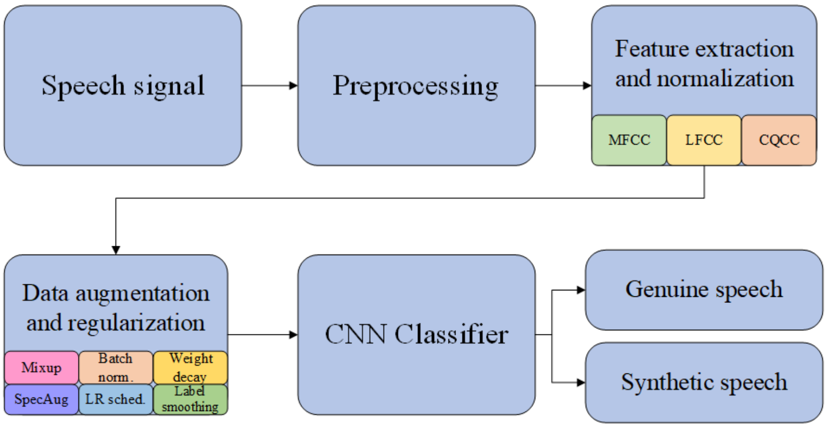

Fig. 1. Block diagram of the proposed synthetic speech detection system.

In the preprocessing stage, the audio recordings from the dataset are first resampled at 16 kHz and converted to single-channel PCM format. The amplitude of each utterance is normalized to the standard [–1, 1] range, silent intervals are

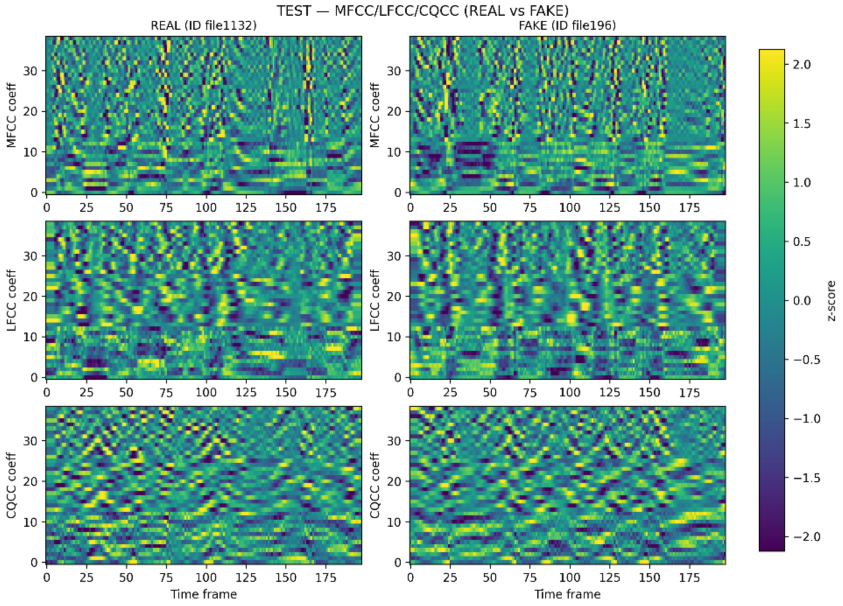

Fig. 2. Cepstral representations of one genuine and one synthetic utterance from the Fake-or-Real dataset. Rows show MFCCs, LFCCs, and CQCCs; columns correspond to the genuine (left) and synthetic (right) utterances. Values are normalized per coefficient and displayed with a single, shared color scale across all panels. Lower MFCC indices highlight spectral envelope / formant dynamics; LFCCs distribute sensitivity more uniformly over frequency; CQCCs capture both envelope and harmonic/periodic structure.

removed, and a 7-point median filter is applied to reduce noise. Each sample is then segmented into 25 ms frames with 60% (15 ms) overlap using Hamming windows prior to feature extraction.

Following data preprocessing, a single cepstral feature set is extracted for each experiment, selected from three alternatives: the first 13 Mel-frequency cepstral coefficients (MFCCs), the first 13 linear frequency cepstral coefficients (LFCCs), or the first 13 constant-Q cepstral coefficients (CQCCs). In all cases, the static coefficients are combined with their delta and delta-delta coefficients (equivalent to first- and second-order derivatives), and then statistical functions (mean and standard deviation) are applied across the frames to summarize temporal variation. The resulting values form the input feature vector for the classifier, with a detailed description of the MFCC extraction process available in our previous work [21-22].

The last step in preparing the input data involves applying *z*-score normalization to each descriptor, standardizing their distributions to zero mean and unit variance. To improve robustness and generalization, the training process incorporates targeted augmentation methods, as well as regularization techniques including batch normalization, label smoothing, and adaptive learning rate scheduling.

The CNN classifier takes the feature vector as input, but the hidden layer structure is more complex, consisting of N_g groups of layers, each group comprising a 1D convolutional layer, a batch normalization layer, a max-pooling layer, and a dropout layer, as illustrated in Fig. 3. The final output feature maps are flattened and fed to a classification head comprising a number of fully-connected layers for additional feature refining. The fully-connected layers also include the dropout mechanism to reduce the risk of overfitting. Although not represented in the figure, the training process further incorporates targeted augmentation and regularization

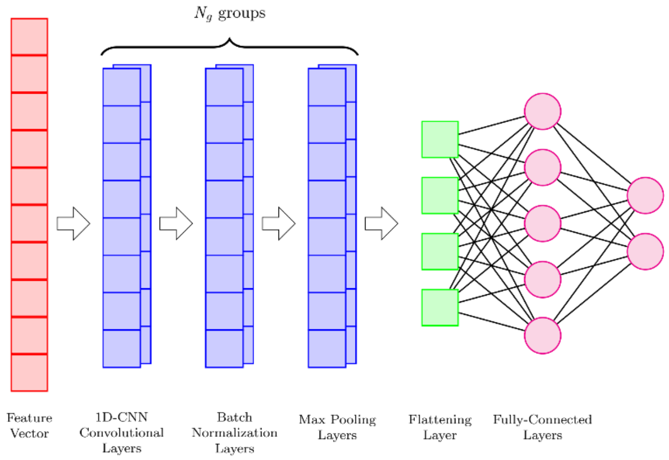

Fig. 3. Architecture of the proposed improved CNN-based synthetic speech detection system.

strategies, including Mixup, SpecAugment, batch normalization, label smoothing, and adaptive learning rate scheduling, to enhance robustness and generalization.

3. Experimental Setup and Results

The experiments were implemented in Python using the TensorFlow framework and were performed on a workstation having an Intel Xeon W-1290P CPU, 128 GB of RAM, and an Nvidia RTX A4000 GPU with 16 GB VRAM.

3.1. The Fake-or-Real dataset

We utilized the publicly available Fake-or-Real (FoR) dataset [4] for our synthetic speech detection experiments. This dataset is explicitly designed for studies in speech synthesis and synthetic speech detection, incorporating the latest developments in deep learning-based Text-to-Speech (TTS) and Voice Conversion (VC) technologies. It provides a comprehensive basis for training complex machine learning models capable of distinguishing synthetic speech from real human speech, and is widely used as a benchmark for the aforementioned tasks.

The complete FoR dataset includes over 111,000 real human utterances and more than 87,000 synthetic utterances produced by 33 distinct speech generation systems. These systems span state-of-the-art open-source solutions like Deep Voice 3 and commercial platforms such as Amazon AWS Polly, Baidu TTS, Google Cloud TTS (both standard and WaveNet versions), Microsoft Azure TTS, and traditional Google TTS. Each synthetic utterance is generated from a diverse set of English phrases, carefully selected to represent a wide range of grammatical structures and sentence complexities, ensuring an important level of generalization capability.

Real speech samples within the dataset were collected from multiple sources to maintain variability in gender, age, accent, and recording conditions, thus preventing potential classification biases. Sources include prominent open-source datasets and high-quality recordings extracted from educational content such as TED Talks and YouTube tutorials.

The FoR dataset is published in four distinct versions: the *for-original* dataset comprising raw files as initially collected; the *for-norm* version, with audio normalized to a consistent WAV format, 16 kHz sample rate, mono-channel, volume normalization, with silence trimming applied, achieving gender and class balance; the *for-2seconds* subset standardized to fixed-length 2-second clips, ensuring audio duration uniformity to eliminate length-based classification bias; and the *for-rerecorded* version, simulating real-world scenarios where synthetic audio was replayed through speakers and re-recorded using common consumer-grade microphones.

We employed the standardized *for-2seconds* subset in our current study due to its balanced representation across gender and class, uniform length, and its practical relevance for realistic classification scenarios. Following the dataset developers' methodology, we partitioned the *for-2seconds* subset into training (77.74%), validation (15.58%), and test sets (6.68%). Crucially, the 3 sets comprise entirely different speakers from each other, both human and synthetic, with the test set including synthetic utterances from Google's WaveNet TTS system, which were deliberately excluded from the training and validation sets to rigorously assess the generalization capability and robustness of the models.

3.2. Experimental setup and details

In our experiments, as shown in Fig. 2, we use cepstral feature vectors with a CNN classifier. Distinct architecture configurations were tested by varying hyperparameters, i.e., the number of layer groups N_g (between 1 and 3), the number of filters for each convolutional layer (8, 16, or 32), the filter kernel sizes (8 or 16), the number of hidden fully-connected layers (between 1 and 3), and the number of neurons in the hidden fully-connected layers (32, 64, 128, or 256). Dropout rates ranged from 0 to 0.5 in increments of 0.05, with experiments also conducted without dropout. During training, we applied targeted augmentation techniques, i.e., Mixup with $\alpha \in \{0, 0.1, 0.2\}$ and SpecAugment with time/frequency masks of varying sizes. Additional regularization strategies included batch normalization (with and without weight decay), and label smoothing ($\varepsilon = 0.1$). Learning stability and convergence were further improved using adaptive learning rate schedules, comparing reduce-on-plateau, cosine decay, and one-cycle policies. All experiments were trained for up to 150 epochs with a batch size of 64, employing early stopping with a patience of 10 epochs based on validation loss.

3.3. Results and discussion

The results obtained for the top 6 MFCC, 6 CQCC, and 4 LFCC configurations, along with their augmentation and regularization setups, are summarized in Table 1. Among all tested combinations, MFCC inputs consistently achieved the highest test accuracies, with the best configuration (Mixup with $\alpha = 0.2$, no SpecAugment, supplemented by label smoothing with $\varepsilon = 0.1$, weight decay of 10^{-4}, and plateau learning rate scheduling) reaching **95.9%** test accuracy and **99.4%** validation accuracy.

Another configuration (SpecAugment with 8/4 time/frequency masking, Mixup with $\alpha = 0.1$, label smoothing with $\varepsilon = 0.1$, weight decay of 10^{-4}, and plateau learning rate scheduling) also achieves the same performance, but involves a higher complexity than the previously mentioned one due to additional augmentation.

Table 1

Performance comparison between types of input feature vectors, data augmentation and regularization techniques

Input feats.	Augmentation & regularization techniques						Val. acc. [%]	Test acc. [%]
	SpecAug. time/freq. masking	Mixup α	Batch norm.	LR sched.	Weight decay	Label smooth. ε		
MFCC	–	0.2	yes	plateau	10^{-4}	0.1	99.4	**95.9**
MFCC	8/4	0.1	yes	plateau	10^{-4}	0.1	99.4	95.9
MFCC	–	0.1	yes	one-cycle	–	–	99.3	95.8
MFCC	–	0.1	yes	plateau	10^{-4}	0.1	99.3	95.8
MFCC	–	0.1	yes	plateau	–	0.1	99.4	93.9
MFCC	10/5	0.1	yes	one-cycle	–	–	99.3	95.5
CQCC	40/20	–	yes	plateau	10^{-4}	0.1	99.4	**90.9**
CQCC	–	0.1	yes	plateau	–	0.1	99.3	90.1
CQCC	–	0.1	yes	plateau	10^{-4}	0.1	99.4	88.1
CQCC	10/5	–	yes	plateau	–	–	99.4	87.8
CQCC	20/10	–	yes	plateau	10^{-4}	0.1	99.4	87.5
CQCC	–	0.2	yes	plateau	10^{-4}	0.1	99.4	87.3
LFCC	10/5	0.1	yes	cosine	–	–	84.0	**65.4**
LFCC	10/5	–	yes	cosine	–	–	83.2	63.6
LFCC	–	0.1	yes	cosine	–	–	83.2	61.4
LFCC	–	0.1	yes	one-cycle	–	–	97.5	60.9

Both configurations maintain a minimal validation-test performance gap, indicating strong generalization without overfitting.

CQCC-based models showed slightly lower peak performance, with the highest test accuracy of 90.9% under light regularization (SpecAugment 40/20, label smoothing with $\varepsilon = 0.1$, plateau learning rate scheduling). Although CQCCs are able to capture in more detail the spectral-temporal structure, their higher variability in test results suggests a greater sensitivity to augmentation settings.

LFCC inputs generally performed the weakest, with the best results peaking at 65.4% test accuracy. These features may discard certain fine-grained phase or harmonic cues needed by the classifier, explaining the large gap in contrast with MFCCs and CQCCs.

Across all feature types, moderate augmentation (small SpecAugment masks, low Mixup α) combined with standard batch normalization provided the best balance between robustness and preservation of discriminative cues. Heavy masking or overly strong regularization consistently widened the validation-test gap and degraded generalization.

Overall, the MFCCs emerged as the most effective standalone input representation for this task on the FoR dataset, with the CQCCs showing competitive but less stable performance. The LFCCs underperformed, highlighting the importance of both the feature extraction method and the regularization strength in synthetic speech detection.

Table 2 presents the best result achieved in this study alongside the top-performing systems from our earlier works. The new augmented CNN-based approach introduced here attains **99.4%** validation accuracy and **95.9%** test accuracy, surpassing both our earlier CNN-based model [1] and our previous augmented transfer learning approach [2].

Compared to [1], which relied solely on CNNs with hand-crafted features and reached 98.9% validation accuracy but only 83.9% on the test set, the present system improves generalization by over 14% on the test data. Against [2], in which we used EfficientNetV2 with spectrogram inputs and attained 97.5% validation and 91.1% test accuracy, the relative improvements are of 1.9% in validation and 5.3% in test performance.

Equally important, the validation-test gap is now just 3.5%, compared to 15% in [1] and 6.4% in [2]. This narrowing demonstrates the robustness of the proposed feature augmentation and regularization combination, which preserves key discriminative cues captured by MFCCs, LFCCs, and CQCCs as inputs while preventing overfitting.

Table 3 compares the best-performing configuration from this work to several recently reported systems in the literature [3-5]. Our augmented CNN approach achieves the highest across all compared methods.

Against the MLP model used in [3], which reached 94.7% validation accuracy, our system achieves an improvement of 4.9%. Compared to the SVM and Random Forest (RF) models proposed in [4], the test accuracy is higher by 22.4% and 24.4% in absolute value, respectively. Even relative to the strong RF baseline in [5] (87.0% test accuracy), our system achieves a relative performance increase of 10.2% while also reducing the validation-test gap from 11.5% to just 3.5%, indicating stronger generalization.

Table 2

Comparison between the best results achieved in this work and our previous work [1-2]

System	Validation accuracy [%]	Test accuracy [%]
Our previous CNN-based approach [1]	98.9	83.9
Our previous augmented transfer learning-based approach [2]	97.5	91.1
Our new augmented CNN-based approach	**99.4**	**95.9**

Table 3

Comparison between the best results achieved in this work and other literature

System	Model	Validation accuracy [%]	Test accuracy [%]
[3]	MLP	94.7	–
[4]	SVM	70.0	73.5
	RF	79.4	71.5
	VGG19	89.8	92.0
[5]	RF	98.5	87.0
This work	**Augmented CNN**	**99.4**	**95.9**

These results highlight that pairing MFCC, LFCC, and CQCC inputs with targeted augmentation and light regularization enables CNN-based systems to surpass both traditional machine learning approaches and more complex architectures while maintaining a relatively low-complexity design and achieving strong generalization to unseen data.

4. Conclusion

In this paper, we presented a synthetic speech detection system based exclusively on conventional convolutional neural networks (CNNs) trained from scratch on cepstral features: Mel-frequency cepstral coefficients (MFCCs), linear frequency cepstral coefficients (LFCCs), or constant-Q cepstral coefficients (CQCCs).

By systematically combining these inputs with targeted augmentation (Mixup, SpecAugment) and regularization techniques (batch normalization, dropout, label smoothing, adaptive learning rate scheduling), we achieved substantial improvements over our previous work and other results presented in recent literature.

On the Fake-or-Real (FoR) benchmark dataset, our system achieved **99.4%** validation accuracy and **95.9%** test accuracy. These results exceed those of our previously proposed CNN-based system, as well as our prior transfer learning-based approach, while also outperforming several recent methods reported in literature. The relatively small gap of 3.5% between validation and test performance highlights our proposed system's generalization capability and the robustness gained from the integration of augmentation and regularization techniques.

Future work will explore hybrid feature representations that combine cepstral and spectrogram-based inputs, cross-dataset evaluation to assess robustness under mismatched conditions, and the extension of the approach to detect partially manipulated speech or more challenging low-quality recordings.

REFERENCES

[1] *I. Mutica*, *S. Mihalache*, *D. Burileanu*, “Synthetic Speech Detection Using Deep Neural Networks,” Proc. International Conference on Telecommunications and Signal Processing (TSP), Brno, Czech Republic, pp. 53-57, Jul. 2024.

[2] *I. Mutica*, *S. Mihalache*, *G. Pop*, *D. Burileanu*, “Augmented Transfer Learning for Synthetic Speech Detection,” Proc. International Conference on Speech Technology and Human-Computer Dialogue (SpeD), Cluj-Napoca, Romania, Oct. 2025. *Accepted for publication*.

[3] *A. Hamza*, *A.R. Javed*, *F. Iqbal*, *N. Kryvinska*, *A. Almadhor*, *Z. Jalil*, *R. Borghol*, “Deepfake Audio Detection via MFCC Features Using Machine Learning,” IEEE Access, vol. 10, pp. 134018-134028, Jan. 2022.

[4] *R. Reimao*, *V. Tzerpos*, “FoR: A Dataset for Synthetic Speech Detection,” Proc. International Conference on Speech Technology and Human-Computer Dialogue (SpeD), Timisoara, Romania, pp. 1–10, Oct. 2019.

[5] *R. Reimao*, *V. Tzerpos*, “Synthetic Speech Detection Using Neural Networks,” Proc. SpeD, Bucharest, Romania, pp. 97–102, Oct. 2021.

[6] *Z. Wu*, *R. K. Das*, *J. Yang* and *H. Li*, “Light Convolutional Neural Network with Feature Genuinization for Detection of Synthetic Speech Attacks,” in Proc. Odyssey: The Speaker and Language Recognition Workshop, Tokyo, Japan, 2020, pp. 288–295.

[7] *Z. Lei*, *Y. Yang*, *C. Liu* and *J. Ye*, “Siamese Convolutional Neural Network Using Gaussian Probability Feature for Spoofing Speech Detection,” in Proc. Interspeech, Shanghai, China, 2020, pp. 1116–1120.

[8] *J. Yang* and *A. Das*, “Long-term High-Frequency Features for Synthetic Speech Detection,” in Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP), Barcelona, Spain, 2020, pp. 6619–6623.

[9] *B. Zhang* and *T. Sim*, “Localizing Fake Segments in Speech,” in Proc. IEEE Int. Conf. Pattern Recognition (ICPR), Montreal, QC, Canada, 2022, pp. 4567–4572.

[10] *D. Salvi*, *P. Bestagini*, *S. Tubaro*, “Reliability Estimation for Synthetic Speech Detection,” Proc. ICASSP, Rhodes Island, Greece, pp. 1-5, Jun. 2023.

[11] *S. Cui*, *B. Huang*, *J. Huang*, *X. Kang*, “Synthetic Speech Detection Based on Local Autoregression and Variance Statistics,” IEEE Signal Processing Letters, vol. 29, pp. 1462-1466, Jun. 2022.

[12] *J. Ye* et al., “One-Class Network Leveraging Spectro-Temporal Features for Generalized Synthetic Speech Detection,” Speech Communication, vol. 169, art. no. 103200, Apr. 2025.

[13] *H. Yang*, *X. Yan*, *H. Wang*, “Dual-Branch Network with Fused Mel Features for Logic-Manipulated Speech Detection,” Applied Acoustics, vol. 222, art. no. 110047, Jun. 2024.

[14] *A. Yadav*, *R. Gupta*, “Compression Robust Synthetic Speech Detection Using Patched Spectrogram Transformer,” Proc. International Conference on Machine Learning and Applications (ICMLA), Florida, USA, pp. 112-118, Dec. 2023.

[15] *Z. Lei* et al., “GMM-ResNet2: Ensemble of Group ResNet Networks for Synthetic Speech Detection,” Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Seoul, South Korea, pp. 12101-12105, Apr. 2024.

[16] *C. Fan* et al., “Spatial Reconstructed Local Attention Res2Net with F0 Subband for Fake Speech Detection,” Neural Networks, vol. 175, art. no. 106320, Jul. 2024.

[17] *A. Firc*, *K. Malinka*, *P. Hanacek*, “Deepfake Speech Detection: A Spectrogram Analysis,” Proc. ACM/SIGAPP Symposium on Applied Computing (SAC), Avila, Spain, pp. 1312-1320, May 2024.

[18] *M. Sahidullah*, *T. Kinnunen*, *C. Hanilçi*, “A Comparison of Features for Synthetic Speech Detection,” Proc. Interspeech, Dresden, Germany, pp. 2087–2091, Sep. 2015.

[19] *M. Todisco*, *X. Wang*, *N. Evans*, “Constant-Q Cepstral Coefficients: A Spoofing Countermeasure for Automatic Speaker Verification,” IEEE Transactions on Information Forensics and Security, vol. 13, no. 5, pp. R-1012–R-1020, May 2018.

[20] *H. Tak*, *J. Patiño*, *A. Nautsch*, *N. Evans*, *M. Todisco*, “An explainability study of the constant-Q cepstral coefficient spoofing countermeasure for automatic speaker verification,” Odyssey Workshop on Speaker and Language Recognition, Tokyo, Japan, pp. 156–163, Dec. 2020.

[21] *S. Mihalache*, *D. Burileanu*, “Using Voice Activity Detection and Deep Neural Networks with Hybrid Speech Feature Extraction for Deceptive Speech Detection,” Sensors, vol. 22, iss. 3, art. no. 1228, Feb. 2022.

[22] *S. Mihalache*, *D. Burileanu*, *C. Burileanu*, “Detecting Psychological Stress from Speech using Deep Neural Networks and Ensemble Classifiers,” Proc. SpeD, Bucharest, Romania, pp. 74-79, Oct. 2021.