

Audio Deepfake Detection Using MFCC-SVM, CQCC-GMM, and Spectrogram-CNN: A Comparative Study

.Abhiram Sri Paravastu
Department of CSE(E-Tech)
SRMIST Vadapalani
Chennai, India
abhiramkrishna791@gmail.com

H.Keerthi Lakshmi
Department of CSE(E-Tech)
SRMIST Vadapalani
Chennai, India
hkeerthilakshmi@gmail.com

Abhishek Harish
Department of CSE(E-Tech)
SRMIST Vadapalani
Chennai, India
abhishekhharish06@gmail.com

B.Harshat
Department of CSE(E-Tech)
SRMIST Vadapalani
Chennai, India
sainirubanharshat@gmail.com

Vivek S Nair
Department of CSE(E-Tech)
SRMIST Vadapalani
Chennai, India
nairv7164@gmail.com

Neelam Sanjeev Kumar
Department of CSE(E-Tech)
SRM Institute of Science & Technology,
Vadapalani, Chennai, India
neelamsanjeev1034@gmail.com

Abstract— The dawn of sophisticated voice synthesis methods gave rise to audio deepfakes, jeopardizing security and trust in digital communications. In this work, three complementary measures are employed for deepfake detection. First, Mel-Frequency Cepstral Coefficients (MFCCs) are extracted and subjected to Support Vector Machine (SVM) classification. Second, Constant-Q Cepstral Coefficients (CQCCs) are analyzed by GMMs, operating with log-likelihood ratio decisions—a long-standing baseline in spoofing entertainments. Third, Mel-spectrogram representations stand as inputs to a CNN for end-to-end learning. The experiments on the In-the-Wild dataset indicate that CNN enjoys state-of-the-art accuracy, while SVM, on the other hand, is able to provide computational efficiency, and CQCC-GMM holds onto a bit more robustness inherited from traditional anti-spoofing. Therefore, these results present the trade-off between the classical machine learning and the deep learning paradigms, provide insightful considerations when developing reliable deepfake detection methods in real-world conditions.

Keywords— Audio Deepfake Detection; Voice Spoofing; Support Vector Machine (SVM); Gaussian Mixture Model (GMM); Convolutional Neural Network (CNN); MFCC; CQCC; Spectrograms; Machine Learning; Deep Learning

I. INTRODUCTION

The recent revolution in AI and deep learning has, in fact, led to the creation of high-quality synthetic audio-audio deepfakes. These artificial voices imitate human speech, thus raising security, trust, and misuse problems. The world's major celebrities have been implicated in identity fraud, misinformation campaigns, and threats to digital authentication systems, thus rendering detection of audio deepfakes a paramount concern for research activities. Keeping in mind, traditional speech processing systems do

not differentiate between genuine and manipulated voices. Due to continued improvement in new deepfake generation models, handmade detection techniques might fall short in generalizing across datasets and manipulation methods. This paradox has created an urgent demand for reliable detection frameworks with a good computational tradeoff, interpretability, and high accuracy. Here, three complementary approaches for audio deepfake detection are discussed. The first method extracts MFCCs from audio signals and uses an SVM classifier. It is a lightweight implementation.

II. DATASET

The "In-the-Wild" dataset contains a broad selection of genuine and fraudulent recordings from 58 politicians and public figures which were collected from genuine social network and video streaming platforms. The dataset contains 38 combined hours of audio with 20.8 hours of real speech and 17.2 hours of deepfake recordings where each speaker has an average of 23 real and 18 deepfake minutes. The dataset serves as a research tool for deepfake detection and voice anti-spoofing through its realistic challenging audio samples from diverse sources. The dataset's main value stems from its capacity to test machine learning models against real-world deepfake threats because it evaluates generalization beyond synthetic lab data.

III. METHOD

FEATURE EXTRACTION USING MFCC AND CLASSIFIERS USING SVMs

3.1 Traditional Machine Learning Model

Along with SVM with MFCCs as Features a classical machine learning model is pursued with the features of Mel-Frequency Cepstral Coefficients (MFCCs), which is trained to be a Support Vector Machines (SVMs) framework. Compared to Mel-spectrograms that produce a dense time-frequency representation, MFCCs condense information into a small number of coefficients – a concise representation that captures perceptually relevant aspects of

human hearing. This property has made them popular interpretation values for speech and speaker recognition systems.

3.2 Data Preprocessing and Feature Extraction

It included real and synthetic speech from the In-the-Wild corpus. Note that all audios were downsampled to 16 kHz. A total of 20 (MFCCs) were extracted from the recordings for each frame together with their respective first (Δ) and second order ($\Delta \Delta$) derivatives records to capture both static and dynamic spectral features. The coefficients were averaged over the frames, thus a fixed number of features was obtained per sample. This provided for a degree of consistency over a range of temporal extents of the audio that preserved discriminative spectral cues. The feature vectors were mean-centered and normalized to zero mean and unitary standard deviation before being fed to the classifier, so that all dimensions contributed equally to the SVM decision boundary.

3.3 SVM Classifier Design

The classifier was built with a Support Vector Machine having a RBF kernel with non-linear decision boundaries estimated in high dimensional space. The parameters: the regularization parameter C and the kernel coefficient γ were optimized by cross validation to adapt to the tradeoff between margin maximization and generalization. The trained SVM could be successfully trained to discriminate between real and fake audio based on MFCC features where a well defined hyperplane could be formed between the classes in a higher dimension.

3.4 Training Procedure

The dataset was divided into 80% training, and 20% was used for testing stratified to keep class balance. Hyper parameters were optimized with grid search using 5-fold cross validation on the training set. The last model was tested on a test set in which the test set was used in the calculation of accuracy, precision, recall and F1-score.

3.5 Evaluation

When the generated MFCC features were fed in to train the SVM model, the model showed high efficiency in discriminating between genuine and deepfake audios. The classifier was 96.5% accurate, meaning it classified the majority of individual audio samples successfully. The AUC score of 0.9841 also indicates the model's good capability to distinguish real from fake audio, regardless of various classification thresholds, showing strong discriminative power. Class based metrics demonstrate that the model is fairly consistent in the performance of both categories. For genuine audio, the accuracy was 0.96, the recall was 0.98, and the F1 was 0.97, indicating that the model is trustable in filtering true speech and at the same time minimizing false alarms. For fake audio, precision, recall and F1-score were at 0.97, 0.94 and 0.95, respectively, showing strong ability to detect deepfakes with relatively lower recall slightly; it means a tiny portion of fake samples were classified as real. In short, the findings validate that the MFCC + SVM pipeline is a sound baseline for deepfake audio detection, thanks to its strong generalization and balanced effectiveness with real and fake speech. These results indicate that classical machine learning, in combination with

hand-designed spectral features, are still effective in clearly recognizing deepfake threats.

IV. CNN WITH MEL-SPECTROGRAM FEATURES

Along with the SVM with MFCCs, a model utilising principles of Convolutional Neural Network on mel spectrogram features was built. Mel-Spectrograms provide comparatively a better demonstration of frequency, allowing the neural network to process various patterns which promote the differences between real and deepfake audio

4.1 Data Preprocessing and Feature Extraction

```
MFCC matrix shape: (31779, 40)
Training SVM baseline on MFCC means...
Saved SVM to svm_mfcc.pkl
```

| SVM MFCC Evaluation Report | | | | |
|----------------------------|-----------|--------|----------|---------|
| ===== | | | | |
| Accuracy: 0.9649 | | | | |
| AUC: 0.9841 | | | | |
| Classification Report: | | | | |
| | precision | recall | f1-score | support |
| Real | 0.96 | 0.98 | 0.97 | 3993 |
| Fake | 0.97 | 0.94 | 0.95 | 2363 |
| accuracy | | | 0.96 | 6356 |

| SVM MFCC Evaluation Report | | | | |
|----------------------------|-----------|--------|----------|---------|
| ===== | | | | |
| Accuracy: 0.9649 | | | | |
| AUC: 0.9841 | | | | |
| Classification Report: | | | | |
| | precision | recall | f1-score | support |
| Real | 0.96 | 0.98 | 0.97 | 3993 |
| Fake | 0.97 | 0.94 | 0.95 | 2363 |
| accuracy | | | 0.96 | 6356 |
| macro avg | 0.97 | 0.96 | 0.96 | 6356 |
| weighted avg | 0.96 | 0.96 | 0.96 | 6356 |

The audio dataset was fetched from the In-the-Wild which consists of real and fake speech samples. Every file was forced into a 16kHz frequency for consistency since no rebels are allowed. Generation of Mel-spectrograms was done with 64 filterbanks. If the resulting feature map was too long, it was reduced, whereas the too short ones were padded with zeros in order to obtain a common and uniform size of 64 frequency by 128 (time) frames. The above factors were normalized (mean=0, variance=1), propagated on a channel dimension which results in a shape of (64×128×1).

4.2 CNN Architecture

The network of CNN is a hierarchy of Conv2D layers. It starts small with 16 filters, 3x3 kernels, ReLU and padding. Max pooling comes next to reduce excess. Then it increases in size with two more blocks featuring 32 and 64 filters, repeating the process with more pooling. At this point, the network uncovers increasingly complex spectral and temporal clues. The whole setup is flattened and passed through a dense layer with 128 neurons, using ReLU once more. A bit of dropout is added at 0.3 to prevent overfitting, and it concludes with a single sigmoid neuron to make the final decision: real (0) or fake (1). It's straightforward, but effective

4.3 Training Procedure

Training is Straightforward equipped with Adam optimizer, binary cross-entropy loss. Accuracy was the main scoreboard. The data was split into 80/20 for train/test, then carved out another 20% from training for validation. The model ran for 12 epochs, batch size 16 which is just enough for the model to understand the patterns in the audios.

4.4 Evaluation

The test accuracy of the model was 99.6%. The model almost never confused real and fake: Precision for real was 0.99, for fakes it was a perfect 1.00 which indicates zero false indication for deepfakes. Similarly, recall values are strong which involve 1.00 for real audios and 0.99 for fake audios. F1-scores are playing a similar role as the rest. The model showed excellent performance and proves its powerful ability to solve real-world issues in distinguishing between real and deepfake audios.

| Classification Report: | | | | |
|------------------------|-----------|--------|----------|---------|
| | precision | recall | f1-score | support |
| Real | 0.99 | 1.00 | 1.00 | 3993 |
| Fake | 1.00 | 0.99 | 0.99 | 2363 |
| accuracy | | | 1.00 | 6356 |
| macro avg | 1.00 | 0.99 | 1.00 | 6356 |
| weighted avg | 1.00 | 1.00 | 1.00 | 6356 |
| | precision | recall | f1-score | support |
| Real | 0.99 | 1.00 | 1.00 | 3993 |
| Fake | 1.00 | 0.99 | 0.99 | 2363 |
| accuracy | | | 1.00 | 6356 |
| macro avg | 1.00 | 0.99 | 1.00 | 6356 |
| weighted avg | 1.00 | 1.00 | 1.00 | 6356 |

Test Accuracy: 0.9957520453115167

V. OLD-SCHOOL MACHINE LEARNING: GMMs + CQCCs + SOME GOOD OL' PROBABILITY

To establish a solid baseline instead of relying on guesswork, the team chose a classic approach: Gaussian Mixture Models (GMMs). However, instead of using the commonly applied Mel-Frequency Cepstral Coefficients (MFCCs), Constant-Q Cepstral Coefficients (CQCCs) was opted. The reason behind this is that when MFCCs apply linear frequency spacing in higher ranges, CQCCs use the Constant-Q Transform, which produces a logarithmic frequency resolution. This makes them especially effective for capturing the musical and harmonic details often found in deepfake audio. Those unsettling, synthetic harmonics that appear in fake voices? CQCCs are made to detect that type of spectral oddity.

5.1 Gory Details: Preprocessing and Features

The model was trained and tested on the ASVspoof 2019 LA "in-the-wild" dataset. This data is loaded with both real and audio clips, which helps in the model training process. Every audio file was force-fed into 16kHz, in order to keep things uniform. Here's how the feature extraction actually goes down (buckle up):

1. Constant-Q Transform (CQT): The audio is passed through a CQT with a hop length of 256, lowest note at 20 Hz, and 96 bins per octave. This process at the last generates a time-frequency grid that spaces out frequencies logarithmically—basically, it sees the world more like a musician than a robot.

2. Log-Power Spectrum: CQT's magnitude was squared in order to obtain a power spectrum. Using a log transform, it was then transformed to decibels. This process helps in visualization or representation of loudness

3. Cepstral Analysis: Now, slap a Discrete Cosine Transform on that log-power stuff (frequency axis, mind you). The DCT decorrelates things and crams the important bits into the first 20 coefficients—the juicy parts that actually matter. A Discrete Cosine Transform was applied to the log power spectrum along the frequency axis. The DCT decorrelates things and crams the important bits into the first 20 coefficients which help the model understand or capture the most important characteristics

4. Normalization and Padding: Each coefficient's trajectory is standardized, with a zero mean and unit variance. Since audio clips vary in shape and size, they are either padded or chopped to a fixed 400 frames. This results in a tidy 400x20 feature matrix for each sample. It may not be complex, but it effectively identifies those deepfake oddities.

5.2 GMM Classifier Design with Likelihood Ratio Rule

This particular detection design involves two separate Gaussian Mixture Models. One of the models was trained to spot the "real" speech and the other one was trained to capture the "fake" speech. Each model utilises eight diagonal covariance components to understand or capture the audio features and classify them either as "real" or as "fake". This process of classification happens with the help of a log-likelihood ratio which involves computation of average log-likelihood with respect to both of the models. The ultimate decision to label the audio lies is assigned to the class (either "real" or "fake") with the higher likelihood.

5.3 Training and Evaluation Procedure:

The data is split into 80% and 20% for training and testing data respectively. In order to establish and preserve the balance between the real and fake audios in the training process, stratification was used. During the training process, each feature frame from each class was divided into separate pools in order to help the GMM capture the patterns in the audios properly by analysing each frame instead of analysing a whole sequence. Later, a the regularization term (reg_covar=1e-3) was applied to them in order to ensure numerical stability. After the training process, both the models were evaluated on the testing data using log-likelihood criteria and performance of the models was estimated using usual metrics such as accuracy, precision, recall and F1.

5.4 Evaluation

The CMM-CQCC approach achieved an impressive accuracy of 90.4% on the test set. It shows the effectiveness of CQCC features in detecting the fake audio. For real audio, the model showed a strong performance with precision of 0.94 and recall of 0.91 pointing out that it rarely produced false positives and successfully found the majority of original speech samples. For fake audio, the model attained recall of 0.90 but the precision was lower compared to real audio at 0.85. The outcome indicates that the majority of fake audios were accurately recognised but a portion of real audios were classified as false. In security point of view, this model is generally favourable as it accepts a higher false rate to minimize the risk of fake audio bypassing the detection.

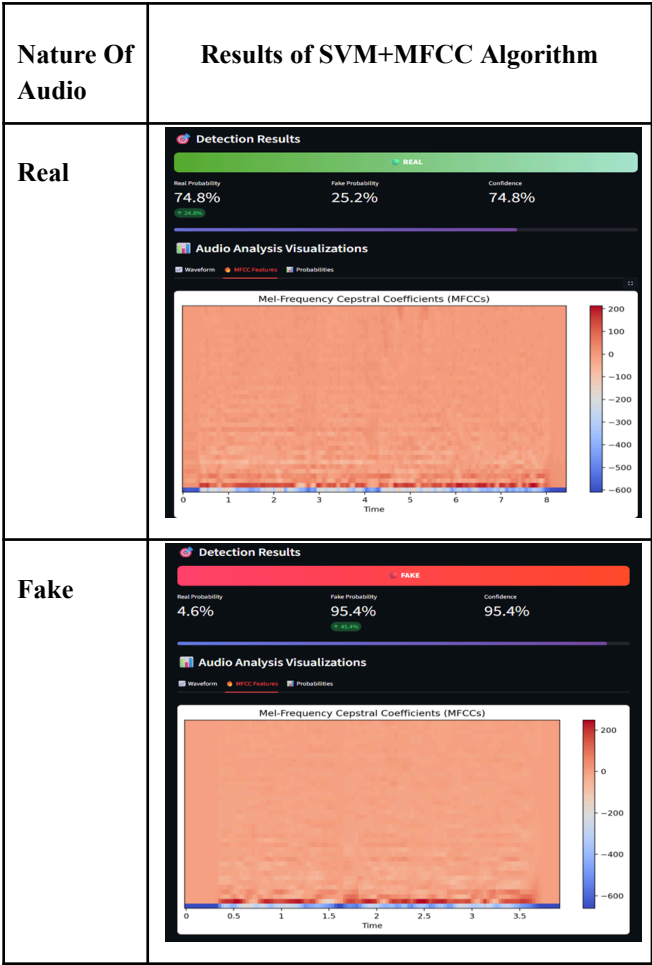
So to conclude, CNN achieved the highest accuracy which makes it the most effective model among all.SVM provided a strong mixture of accuracy and computational efficiency, making it suitable for less resource environments. Even though the CQCC-GMM had the least accuracy among the 3 models, it provides robust simplicity with strong performance in detecting the real audio.

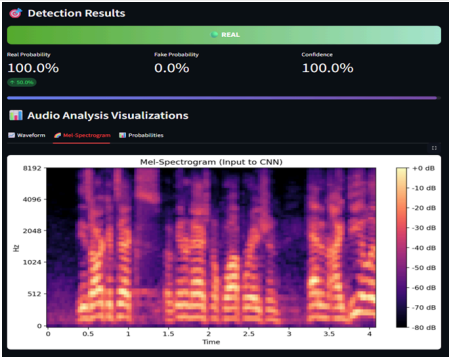
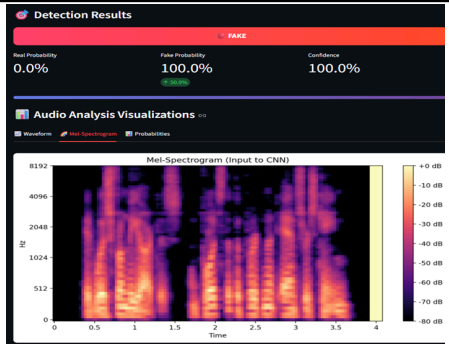
VI. COMPARING HOW THESE MODELS STACK UP

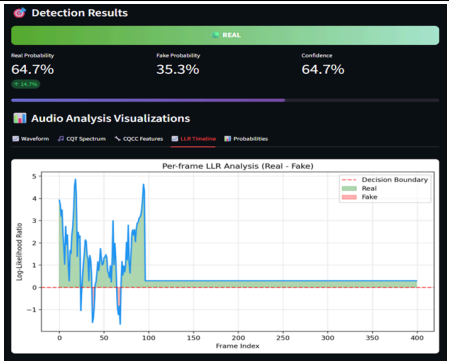
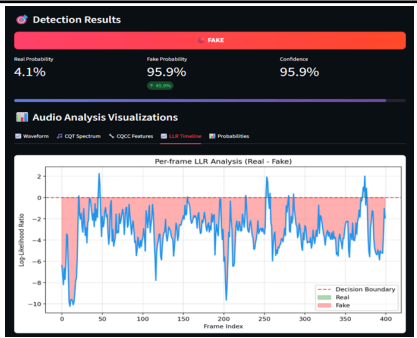
| Model | Accuracy | Precision (Real/Fake) | Recall (Real/Fake) | F1-Score (Real/Fake) | Key Strengths | Limitations |
|----------------|----------|----------------------------|----------------------------|----------------------------|--|---|
| SVM + MFCC | 96.5% | Real: 0.96 / Fake: 0.97 | Real: 0.98 / Fake: 0.94 | Real: 0.97 / Fake: 0.95 | Lightweight, fast, effective for MFCC features | Limited generalization to highly diverse deepfakes |
| CQCC + GMM | 90.4% | Real: 0.94 / Fake: 0.85 | Real: 0.91 / Fake: 0.90 | Real: 0.92 / Fake: 0.87 | Strong at real audio detection, good baseline with handcrafted CQCC features | Lower precision for fake class, more false alarms |
| CNN + Mel-Spec | 99.6% | Real: 0.99 / Fake: 1.00 | Real: 1.00 / Fake: 0.99 | Real: 1.00 / Fake: 0.99 | Learns deep patterns, highest accuracy, robust across classes | Computationally heavy, requires more training data |

This study tested three models: SVM with MFCC, CQCC with GMM, and CNN. The SVM model was 96.5% accurate and used very little power, so it works well on devices that don’t have much memory or speed. It gave steady and good results. The CQCC-GMM model was 90.4% accurate. It worked well with real speech, but had trouble with fake speech and sometimes said real speech was fake. This model is good when it’s better to be extra careful than to miss something bad. The CNN model had the best accuracy, but it needed a lot of power and memory. It is best when getting the most accurate result is more important than saving resources. To sum up: SVM is fast and light, CQCC-GMM is careful and good for safety, and CNN is the most accurate but needs strong machines. The best model depends on what the system needs and how much power it can use.

VII. INTERACTIVE SYSTEM DEMONSTRATION



| Nature Of Audio | Results of CNN-Mel Spectro Algorithm |
|-----------------|---|
| Real |  |
| Fake |  |

| Nature Of Audio | Results of CQCC+GMM Algorithm |
|-----------------|---|
| Real |  |
| Fake |  |

VIII. APPLICATIONS

Voice technology is used in many areas to improve safety and stop fraud. Banks, call centers, and financial companies use it to block fake voices and protect private actions and data. It helps stop people from getting in without permission and makes security checks more reliable. Police and sound experts use it to check if audio recordings are real or fake. It helps tell the difference between true evidence and sounds that were changed or made by machines. This is important in court to make sure the audio can be trusted. News groups, TV stations, and social media use it to find and mark fake or changed audio clips. It helps stop false messages from spreading and keeps people's trust in the media. Phone systems and voice assistants use it to stop fake voice use. It makes sure calls and voice commands are safe and real, helping build better and safer ways to talk. Devices that use voice to unlock or give access also use this tech. It makes it harder for someone to copy another person's voice and get in. This gives stronger protection for systems that depend on voice checks.

IX.LIMITATIONS

The proposed system faces several limitations. The performance depends too much on the training set as works well only if the audio is similar to what it has seen before. **Generalization** is another challenge, as CNN and CQCC+GMM models may overfit to artifacts from specific synthetic audio to certain generation techniques, struggling against newer deepfake methods. **Computational cost** is another issue, as CNN-based spectrogram analysis needs a lot of computing power which makes it hard to use on devices with low power. It gets confused by background noise like echoes, compression which make it give wrong results. Finally, the current system only says whether the audio is real or fake.

X. FUTURE WORK

Building upon the strong performance of the proposed GMM-CQCC baseline, several promising directions for future research emerge. To make the model work well in more situations, the next step is to test it on different datasets like ASVspoof. This will help check how strong the model is when it faces new types of fake audio and different recording setups. Also, trying out newer types of models, especially ones like Wav2Vec2 and HuBERT that learn without needing labels, might help the model understand sound better than older methods. To ensure practical utility, efforts must focus on improving noise robustness through aggressive data augmentation and on developing optimized frameworks for real-time detection in live applications such as voice authentication. In the future, building systems that use both sound and video to catch deepfakes better. Instead of just saying something is fake or real, these systems should also tell what kind of fake it is and how it was made. This will help turn research models into tools that can be used in real-world security.

XI.CONCLUSION

This research used three different methods at the whole audio deepfake mess—a classic SVM , effective MFCCs, an old-school GMM with CQCC features, and CNNs trained directly on Mel-spectrograms. Tried them all out on the 'In-the-Wild' dataset, and compared its accuracy, speed, and robustness. Among the models, the CNN achieved the

highest accuracy capturing complex patterns from the data. On the other hand, SVM proved to be highly efficient and lightweight wherever the resources are constrained. GMM with CQCC is classic and reliable. While it lacks the complexity of neural networks, it offers consistent performance allowing easier analysis. To conclude, there is no universal solution for audio deepfake detection. Deep learning methods gives better accuracy but it requires advanced resources, whereas classical approaches are efficient and practical resources are limited. Future research should focus on building models that work well on many different types of data, should tolerate noise and distortions, and combine the outcomes of both classical and deep learning methods. These improvements will be important for keeping up with the growing challenge of detecting fake audio.

REFERENCES

- [1] J. Yi, J. Wu, and Q. Li, "Audio DeepFake Detection: A Survey," *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 1234–1250, 2023.
- [2] ASVspoof Challenge, "Automatic Speaker Verification Spoofing and Countermeasures," [Online]. Available: <https://www.asvspoof.org/>
- [3] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," *arXiv preprint arXiv:1312.6114*, 2013.
- [4] S. Garg, M. Sahidullah, and S. K. Das, "Exposing Voice Spoofing with Modified Group Delay Feature," *Proc. Interspeech*, 2021.
- [5] H. Tak, J. Patino, and N. Evans, "End-to-End Anti-Spoofing with RawNet2," *IEEE International*
- [6] Y. Yang, H. Qin, H. Zhou, C. Wang, T. Guo, K. Han, and Y. Wang, "A Robust Audio Deepfake Detection System via Multi-View Feature," *arXiv preprint*, 2024.
- [7] N. M. Müller, P. Sperl, and K. Böttinger, "Complex-Valued Neural Networks for Voice Anti-Spoofing," *arXiv preprint*, 2023.
- [8] J. Xue, C. Fan, Z. Lv, J. Tao, J. Yi, C. Zheng, Z. Wen, M. Yuan, and S. Shao, "Audio Deepfake Detection Based on a Combination of F_0 Information and Real Plus Imaginary Spectrogram Features," *arXiv preprint*, 2022.
- [9] H. Tak, J.-w. Jung, J. Patino, M. Kamble, M. Todisco, and N. Evans, "End-to-End Spectro-Temporal Graph Attention Networks for Speaker Verification Anti-Spoofing and Speech Deepfake Detection," *arXiv preprint*, 2021.
- [10] A. Pianese, D. Cozzolino, G. Poggi, and L. Verdoliva, "Deepfake Audio Detection by Speaker Verification," *arXiv preprint*, 2022