Journal of Posthumanism
2025
Volume: 5, No: 8, pp. 341–369
ISSN: 2634-3576 (Print) | ISSN 2634-3584 (Online)
posthumanism.co.uk

DOI: https://doi.org/10.63332/joph.v5i8.3164

# The Impact of Data Distribution and Feature Selection on Machine Learning Performance in Fake Audio Detection

Asmaa M. S. Abo Alfadl[1], Mohamed H. Khafagy[2], Engy R. Abdelmaksoud[3], Ahmed S. Ismail[4]

***Abstract***

*The aim behind this research is to investigate the effect of using machine learning algorithms in enhancing the performance of fake audio detection, particularly after applying different data distribution patterns and feature selection techniques. The study is conducted under three data distribution scenarios: a balanced dataset with equal real and synthetic samples, a real-dominant dataset, and a fake-dominant dataset. Three classification algorithms (RFC, SVM, and GB) are implemented to analyze the performance of audio features with different sizes across the three classifiers. The experimental data revealed that RFC and GB achieved better performance compared to SVM by reaching 99% accuracy in balanced conditions. The research obtained its datasets from two different sources: the original clean CFAD dataset and the rerecorded "For-rerec" version of the Fake-or-Real dataset. The results indicate that data distribution, in conjunction with feature richness, plays a crucial role in developing dependable fake audio detection systems.*

***Keywords:*** *Audio Deepfakes, Artificial Intelligence, Deep Fakes, Feature Selection, Fake Audio Detection, Machine Learning.*

## Introduction

There has been significant development in artificial intelligence (AI) technology, which can produce highly realistic human-sounding voices in recent times (Lyu, 2020). Although these technologies were designed to serve humanity, for instance, in audiobooks and assistive technology, they have also been exploited for nefarious purposes. The malicious application includes spreading disinformation worldwide with audio (Nicholas Diakopoulos, 2020), something that raised alarms over the possibility of "Audio Deepfakes" (AD). These sound manipulation methods, which are now readily available via simple mobile phones and personal computers (Yohanna Rodríguez-Ortega, 2020), have caused global cybersecurity alerts regarding their impact on online trust. Unlike standard digital attacks, e.g., false emails or false text links, audio deepfakes attack the voice of a man by employing a compelling and emotive channel, hence becoming more credible and dangerous (Tianxiang Chen, 2020). This technology can also be used as a method of voice spoofing to manipulate public opinion for the purpose of propaganda, defamation, or even terrorism. Due to the large number of voice recordings being broadcast every day through the internet, it has become incredibly hard to identify impersonated audio (Dora M. Ballesteros, 2021). In 2019, AI-based software was used by scammers to imitate

[1] Department of Information Systems, Faculty of Computers and Artificial Intelligence, Fayoum University, Fayoum 63514, Egypt, Email: am4635@fayoum.edu.eg, (Corresponding Author)
[2] Department of Computer Science, Faculty of Computers and Artificial Intelligence, Fayoum University, Fayoum, Egypt.
[3] Department of Basic Science, Faculty of Computers and Artificial Intelligence, Fayoum University, Fayoum, Egypt
[4] Department of Information Systems, Faculty of Computers and Artificial Intelligence, Fayoum University, Fayoum 63514, Egypt

the voice of a CEO and fraudulently trick the company out of over USD 243,000 through a phone call (Stupp, 2019). Therefore, it has become essential to authenticate any audio recordings so that misinformation is not spread. Therefore, this issue has been given significant attention within the research community for the past few years. The rate at which deep learning and artificial intelligence have developed has significantly transformed the field of audio synthesis, and this has created highly realistic fake audio, commonly referred to as audio deepfakes. Files of synthesized audio can so accurately impersonate human speech that they have created substantial security, privacy, and authenticity issues in digital communication. As the technology advances, it becomes increasingly difficult for individuals and systems to identify real and created speech. Artificial speech deepfake detection is, therefore, a pressing and important research area. Its principal goal is to develop approaches that can reliably differentiate between genuine and manipulated or synthesized speech. However, the ever-increasing sophistication of modern synthesis techniques, most notably those based on Generative Adversarial Networks (GANs), auto encoders, and Transformer models, has made detection a very complex process. These models can synthesize speech that is nearly indistinguishable from real recordings, not only in the sense of what is spoken but also in speaker identity, prosody, and acoustic detail. There is no concrete security mechanism yet to ensure secure and verified access for users, despite the revolutionary potential of today's internet technologies. This weakness is further amplified in biometric systems, such as voice authentication, in which deepfakes can be employed in an attempt to bypass identity validation measures. Such loopholes emphasize the need for effective deepfake detection systems with capabilities to match the rapid pace of technology development. The structure of the training dataset is one of the major factors influencing the performance of models detecting fake audio. Data imbalance, when one class (real or simulated audio) dominates the other, can significantly skew model performance, reduce generalization, and result in skewed predictions. To mitigate this, this paper investigates three data-first scenarios: the first scenario, equal numbers of real and simulated audio samples; the second scenario, higher numbers of real than simulated samples; and the third scenario, higher numbers of fabricated samples than real. This comparative approach is sought to provide a comprehensive insight into the influence of class distribution on model accuracy as well as stability. This study employs a wide range of feature extraction techniques and analyses the spectral and temporal as well as cepstral features of the sound signals. Feature sets ranging between 20 and 160 features were employed to train several machine learning classifiers, including Random Forest (RFC), Support Vector Machine (SVM), and Gradient Boosting (GB). Experimental results indicate that RFC and GB perform better than SVM consistently, particularly in balanced or mildly imbalanced datasets, with accuracy levels as high as 99%. There are, nevertheless, still some challenges in this area. These include access to large-scale, varied, and public datasets, real-time processing requirements, and the ability to generalize across languages, accents, speaker qualities, and recording conditions. Furthermore, the continually evolving synthesis methods require adaptive and forward-looking detection paradigms. Latest efforts suggest integrating artificial intelligence with multimodal and hybrid methods can offer potential solutions to these problems and enhance subsequent detection systems' robustness.

## A. Deepfake and Audio Deepfake Threats

Deepfakes can be described as artificial information or content created or altered using artificial intelligence (AI) technologies and are intended to be believed as actual. These include examples such as audio, video, image, and text synthesis (Zahra Khanjani, 2021) . More specifically, the

term "deepfake" is utilized with the integration of the terms "deep learning" and "fake," with advances in artificial neural networks (ANNs) being utilized to alter media content. Common consumer usage, such as FaceApp and FakeApp, has been used to superimpose someone's face onto other individuals' videos, thereby creating false or entirely false scenarios. With these systems, a person can easily alter his or her appearance, age, or even hair, and bring about a host of problems with the spread of digital fakes and eroded confidence in messages from the media. Deepfake technology has recently also spread its wings to the audio world, producing what has come to be referred to as Audio Deepfakes. These are impersonation audio clips that can mimic a human voice very convincingly, with voice pattern, tone, and personality. Though designed initially for beneficial applications, such as assistive technology and entertainment software, audio deepfakes have been misused to distribute misinformation and conduct social engineering attacks. Their increasing prevalence via smartphones and desktop computers has stimulated widespread public concern for their potential cybersecurity impact. From their perspective, audio deepfakes are more problematic than other conventional digital attacks, such as spam emails or phishing links, since they exploit the human voice, a psychologically persuasive and emotionally compelling medium that is more sinister and more challenging to detect (Tianxiang Chen, 2020) They can be used for propaganda, defamation, or even terrorism, by providing the impression that individuals said something they never spoke. Detection of manipulated content among the vast number of sound recordings shared daily on digital platforms remains a significant challenge. Moreover, political figures and state governments are also being targeted by deepfake attacks (Supasorn Suwajanakorn, 2017) .In a 2019 case, for example, fraudsters used AI software to impersonate a company's CEO's voice on a phone call and were able to dupe the organization out of more than $243,000. Since such threats are so severe and so sophisticated, especially in the audio sector, the need for proper detection systems that can verify the origin of digital audio has become a critical issue in artificial intelligence research and cybersecurity. In view of that, any audio recordings being made publicly available should be verified for authenticity to prevent the spread of misinformation. Therefore, this topic is of interest to the scientific community at the present time. It is getting more and more difficult to detect audio forgery as a result of the creation of three various forms of deepfakes, such as synthetic data-based, imitation data-based, and replay-based.

## B. Types Of Audio Deepfake Attacks

AD technology is a recent invention that allows users to create audio clips that sound like specific people saying things they did not say. This technology was initially developed for a variety of applications intended to improve human life, such as audiobooks, where it could be used to imitate a soothing voice (Anupama Chadha, 2021). As defined from the AD literature, there are three main types of audio fakeness: imitation-based, synthetic-based, and replay-based Deepfakes.

**1) Imitation-based Deepfakes**

Imitation-based Deepfakes is a way of transforming speech (secret audio) so that it sounds like another speech (target audio) with the primary purpose of protecting the privacy of the secret audio" (Yohanna Rodríguez-Ortega, 2020). Voices can be imitated in different ways, for example, by using humans with similar voices who can imitate the original speaker. However, masking algorithms, such as Efficient Wavelet Mask (EWM), have been introduced to imitate audio and Deepfake speech. In particular, the original and target audio will be recorded with similar characteristics. Then, as illustrated in Figure 1, the signal of the original audio that is shown in a Figure 1 – (a) will be transformed to say the speech in the target audio in Figure 1 – (b) using an imitation generation method that will generate a new speech, shown in Figure 1-(c), which is the fake one. It is thus difficult for humans to discern between the fake and real audio generated by this method.

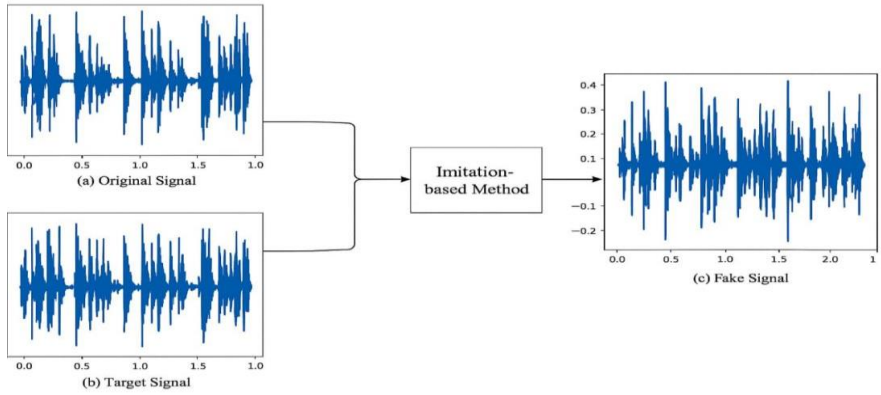


Figure 1. Imitation-based Deepfake

**2) Synthetic-based or Text-To-Speech (TTS)** aims to transform text into acceptable and natural speech in real time and consists of three modules: a text analysis model, an acoustic model, and a vocoder. To generate synthetic Deepfake audio, two crucial steps should be followed. First, clean and structured raw audio should be collected, with a transcript of the audio speech. Second, the TTS model must be trained using the collected data to build a synthetic audio generation model. Tactoran 2, Deep Voice 3, and FastSpeech 2 are well-known model generation techniques and are able to produce the highest level of natural-sounding audio. Tactoran 2 creates Mel-spectrograms with a modified WaveNet vocoder (Jonathan Shen, 2017). Deep Voice 3 is a neural text-to-speech model that uses a position-augmented attention mechanism for an attention-based decode (Wei Ping, 2017). FastSpeech 2 produces high-quality results with the fastest training time. In the synthetic technique, the transcript text with the voice of the target speaker will be fed into the generation model. The text analysis module then processes the incoming text and converts it into linguistic characteristics. Then, the acoustic module extracts the parameters of the target speaker from the dataset depending on the linguistic features generated from the text analysis module. Last, the vocoder will learn to create speech waveforms based on feature parameters, and the final audio file will be generated, which includes the synthetic fake audio in a waveform format. Figure 2 illustrates the process of synthetic-based voice generation.

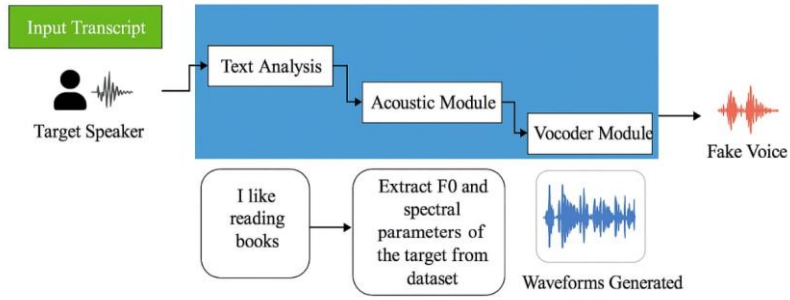


Figure 2. The Synthetic-based Deepfake Process.

**3) Replay-based Deepfakes** are a type of malicious work that aims to replay a recording of the target speaker's voice (Zahra Khanjani, 2021). There are two types: far-field detection and cut-and-paste detection. In far-field detection, a microphone recording of the victim is played as a test segment on a telephone handset with a loudspeaker. Meloudspeakerutting and pasting involves faking the sentence required by a text-dependent system (Swadhin Pradhan, 2019).

The rest of research is organized as the following; where the Literature Review is demonstrated in section2, while the proposed methodology is illustrated in section 3, and the results is represented and discussed in section 4.

## Literature Review

The introduction of deepfake audio technology, enabled by advanced generative models such as GANs and neural TTS systems, has been fueled by mounting interest in developing robust detection techniques. It differs from traditional machine learning techniques to push deep learning systems to the next level, typically evaluated with benchmarking corpora such as ASVspoof and Fake-or-Real (FoR) (J. Ameer Hamza, 2022) Investigated deepfake audio detection using MFCCs as the main feature set in both machine learning and deep learning pipelines. Utilizing the FoR dataset, which is audio produced by state-of-the-art TTS models, they found that Support Vector Machine (SVM) worked best on short-length samples and VGG-16 on full-length audio. Similarly, analyzed feature-based and image-based approaches using Mel-spectrograms and employed models such as Temporal Convolutional Networks (TCNs) and Spatial Transformer Networks (STNs). The research highlighted the effectiveness of time-domain representation and established that TCNs achieved a maximum accuracy of 92%. Authors in (Dora M. Ballesteros, 2021) presented Deep4SNet, a CNN-based classifier that transforms audio into histograms. Their imitation voice identification model and Deep Voice systems achieved 98.5% accuracy. While authors in (Mvelo Mcubaa, 2023) extensively experimented with some of the CNN-based models, including VGG-16, ResNet, and custom models, on different representations of audio (MFCCs, spectrograms, chromagrams). Their results indicated that a light-weight custom CNN operated best overall, but VGG-16 particularly operated well using MFCCs. In an even more targeted linguistic environment, (ELGIBREEN, 2023) developed the Arabic-AD system to detect deepfake audio in Modern Standard Arabic. Using self-supervised learning (SSL), they came up with the first Arabic-language deepfake dataset, consisting of accented and non-accented speakers. Their research emphasized the impact

of speaker diversity and accent variation on detection performance. Apart from that, earlier research has contrasted a variety of spectral features, MFCCs, spectrograms, and chromagrams, to their applicability in spoof detection. Research has proven that employing ensemble-based audio features can dramatically improve detection performance, especially when combined with classifiers like GMM-UBM or SVM. CNN-RNN models have, in general, outperformed traditional approaches like HMMs due to their improved ability to learn temporal dependencies in speech. The FoR dataset itself was introduced as a large benchmark for synthetic speech detection with over 198,000 utterances across real and synthetic speech. It has already been used to evaluate model performance and shown that VGG-16 outperforms other models when trained on the data. In contrast, Temporal Convolutional Networks (TCNs) are particularly effective in the identification of long-range dependencies in time-series data in outperforming LSTMs and MLPs in audio deepfake detection tasks. Though progress has been made, much of the earlier work supposes class-balanced datasets, ignoring real-world situations where class imbalance is the norm (e.g., prevalence of either real or fabricated audio).

This reveals a significant deficiency in the understanding of model behavior under such settings. This work addresses this with an empirical comparison of machine learning models under balanced, real-dominant, and fake-dominant class distributions. It also investigates how feature dimensionality impacts classification performance, yielding greater insights into the detection models' generalization and robustness in real-world scenarios. Imposter speech attacks have become a major threat to automatic speaker verification (ASV) systems. While most detection methods currently have favorable performance in a given dataset, they rarely generalize to new spoofing attacks. Fine-tuning and retraining are the conventional methods to address this limitation; however, fine-tuning can lead to catastrophic forgetting, and retraining is computationally costly and thus impractical in most applications for reasons of privacy.

To mitigate these challenges, (Haoxin Ma J. Y., 2021) proposed a continual learning-based method called "Detecting Fake without Forgetting", which allows for incremental learning of new spoofing attacks without sacrificing performance on seen attacks. Their approach incorporates a knowledge distillation loss to preserve prior knowledge and an embedding similarity loss to preserve alignment of real voice features under the assumption of distributional consistency. Experimental results on the ASVspoof2019 corpus showed that their approach outperformed typical fine-tuning, yielding a relative reduction in Equal Error Rate (EER) of as much as 81.62%. Audio Deepfakes (ADs) have become more dangerous as voice cloning techniques have been misused. Although they were intended for beneficial purposes, such as audiobooks, they were exploited to pose a risk to public security. Therefore, various Machine Learning (ML) and Deep Learning (DL) methods have been proposed to combat such attacks. (Zaynab Almutairi, 2022) has conducted an in-depth analysis of modern AD detection methodologies. Their contribution classifies AD attacks into imitation-based and synthesis-based attacks and provides a comparative analysis of the detection mechanisms and datasets available. The authors observe that performance is more influenced by the detection mechanism type compared to the audio feature set, with a clear tradeoff between scalability and accuracy. Moreover, the review also covers robustness concerns of real-world robustness, mainly in the case of accented speech and background noise, and covers significant future work areas. In doing so, it makes the work a fundamental reference point for understanding the state-of-the-art in AD detection.

## 1. Proposed Methodology

The main goal behind the research methodology is to detect highly accurate fake audio. Initially, the dataset undergoes a preprocessing stage to ensure quality and consistency, followed by feature extraction to capture important acoustic and spectral characteristics. Afterward, feature selection techniques are applied to identify the most relevant features for classification.

The dataset is then divided into training and testing subsets through a sampling process. Three machine learning models, XGBoost, Support Vector Machine (SVM), and Random Forest Classifier (RFC), are trained using the training dataset. The trained models are evaluated on the test dataset, and the detection process classifies the audio as either real or fake. Finally, the performance of each model is assessed using key evaluation metrics, including accuracy, precision, recall, and F1-score, to determine the most effective approach for fake audio detection, as shown in Figure 3.

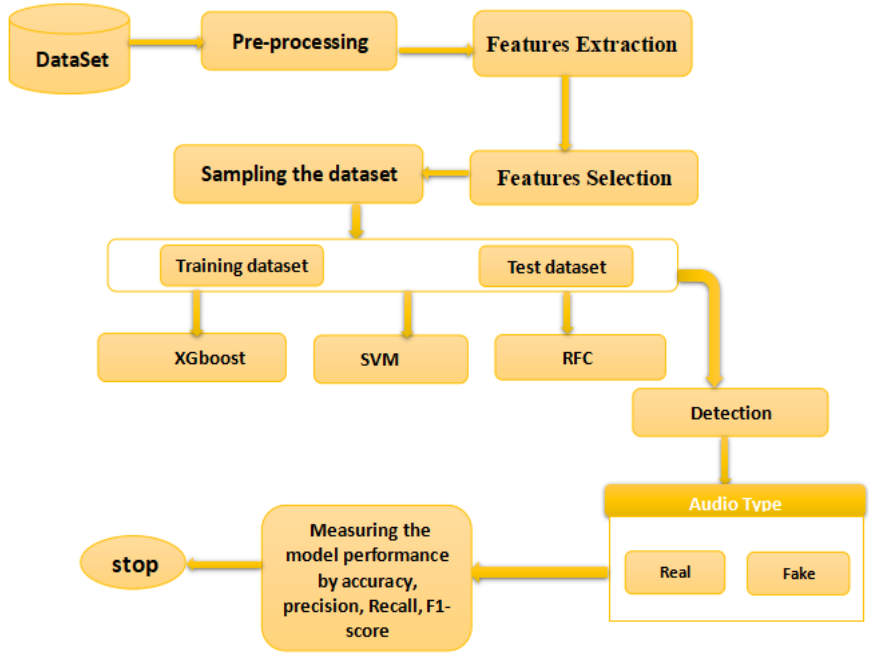


Figure 3. The illustration of The Impact of Data Distribution and Feature Selection on Machine Learning Performance in Fake Audio Detection

### A. Dataset

In this study, we have used Fake-or-Real and CFAD datasets containing various synthetic speech samples. Use the cleaned data version. We used the Fake-or-Real dataset, which is the most recent benchmark dataset. The dataset was created with a text-to-speech model and the CFAD dataset, which considers 12 types of fake audio, 11 of which are generated by different speech synthesis techniques, and the remaining one is a partially fake type.

1) **The FoR dataset** (Ricardo Reimao, 2019) is a set of around 195,000 utterances from real humans and computer-generated speech. The computer-generated samples are created using

the latest methodologies in speech synthesis. The real utterances have samples recorded using different microphones that include different samples from a variety of accents and genders. Classifiers can be trained using this dataset to identify synthetic speech. There are four versions of this dataset such as for-orig, for-norm, for-2sec, and for-rerec

a) **For original***:* It consists of a collection of original files from various speakers. This dataset has an unequal number of samples from different genders. It has around 195,541 samples from various sources, and it was published so that the raw data can be used in various pre-processing techniques.

b) **For-norm:** It has similar files to those of for-original, and it has an equal number of samples from different genders and classes. It is normalized in terms of volume, sample rate, and several channels. The for-norm version consists of around 69,400 samples.

c) *For-2sec:* This dataset consists of files trimmed to 2 seconds. It has an equal number of samples from different genders and classes, just like the normalized version (for-norm), and has a total of 17,870 samples from these sources.

d) **For-recrec:** This version of the dataset includes rerecorded files from the for-2sec dataset. This is done to simulate a situation where an audio is sent through a phone call or voice message, or a similar voice channel, by any attacker.

2) **CFAD Dataset** (Haoxin Ma ,. J., 2023) in recent years; the research community has increasingly recognized that fake audio attacks occurring in real-life scenarios demand serious attention. While early datasets such as those used in the ASVspoof Challenges focused primarily on spoofing within automatic speaker verification (ASV) systems, the ASVspoof 2021 Challenge extended its scope to address deepfake detection beyond ASV**,** reflecting the broader threat posed by synthetic speech in diverse applications. To support the growing need for more realistic and varied evaluation conditions, several new datasets have been introduced, including FoR**,** WaveFake**,** HAD**,** and FMFCC-A**.** These resources have greatly facilitated progress in fake audio detection. However, existing detection models still suffer from poor generalizability, particularly when tested on out-of-domain or unseen scenarios. Real-world audio often contains various types of noise, compression artefacts, and language-specific variations that are not adequately covered in many existing datasets, most of which are English-centric. To address these limitations, the CFAD (Chinese Fake Audio Detection) dataset was introduced. Although primarily designed for Mandarin, CFAD presents a rich and diverse tested for evaluating model generalization**,** robustness, and realism. The dataset includes 12 types of fake audio**,** 11 of which are generated using different synthesis techniques, and 1 category of partially fake audio**,** which is distinct from fully synthetic speech and useful for testing generalization to unknown manipulations. Real audio samples collected from six different corpora, enhancing the diversity and reducing dataset-specific bias. Simulated real-world condition**s** with varying noise types**,** media codecs**,** and signal-to-noise ratios (SNR)**.** Detailed labels indicating fake type, real source, noise condition, and codec information. CFAD is structured into three versions: Clean (unaltered audio), Noisy (audio with background interference), and Codec (compressed audio simulating media conditions). By incorporating diverse audio conditions and comprehensive annotation, the CFAD dataset provides an essential platform for developing and evaluating fake audio detection systems that can generalize well across languages**,** noise conditions**,** and unknown synthesis methods. Its use in this study supports rigorous benchmarking and aligns with the research goal of building robust and adaptive fake audio detection frameworks.

## B. Data Pre-processing

In this work, we used the for-rerec version of the dataset to simulate real-world attack scenarios so that audio files may have been altered during transmission over various communication channels. This allowed us to evaluate the robustness of our model against real-world distortions. The clean CFAD dataset was also utilized. This release contains uncompressed audio samples free from compression artifacts and background noise, allowing detection models to be tested in ideal conditions. The clean set comprises 12 types of artificially generated audio, 11 of which are synthesized using diverse speech synthesis techniques, and one partially fake audio type, which is highly useful in determining whether the model can generalize to unseen manipulation patterns. The real audio samples in CFAD are drawn from six Chinese corpora, which are varied concerning speaker features, recording environments, and speech styles. This variety reduces dataset-specific bias and facilitates training and testing machine learning models. CFAD is built for Mandarin, but the design and labelling are appropriate for testing the basic performance of fake audio detection models. By employing the clean version, the experiment will construct a controlled experimental environment to study the impact of changing feature sets and data distributions without external interference. It is a good foundation before incorporating more advanced audio conditions, such as noise or codec variability, in later work.

## C. Feature Extraction

In the feature-based classification approach, the audio file is converted into a feature-based dataset consisting of various features of the audio samples. First, the individual audio file is taken as input and loaded using the librosa library. This outputs a time series and the sampling rate, which represent the digital form of the audio file. Using the librosa library functions, various features are extracted, including MFCCs (Mel-frequency cepstral coefficients), Mel spectrogram, Chroma, Spectral Centroid, Spectral Bandwidth, Spectral Contrast, Zero Crossing Rate, RMS (Root Mean Square), and *Spectral Roll-off*. For each feature, the mean across the time dimension is calculated to obtain a single value that represents the feature for that particular audio file. These extracted features are then combined into a vector of features per audio sample. This vector is subsequently fed into machine learning algorithms for the classification of audio samples as real or fake. These features were selected because they provide a comprehensive analysis of both temporal and spectral characteristics of sound. Together, they offer a robust framework for understanding the frequency, timbre, harmonic structure, energy, and dynamics of an audio signal, making them highly suitable for applications like speech recognition, music analysis, and environmental sound classification.

## D. Feature Selection

Feature selection is a crucial step in the development of effective machine learning models, particularly in fake audio detection. The selection of the right set of features enables the model to focus on the most informative and relevant characteristics of the audio signal, which can successfully enhance classification accuracy, prevent overfitting, and promote enhanced generalization, especially when dealing with imbalanced or complex datasets. In this study, different sets of audio features were used to examine their effect on the classification accuracy of three machine learning classifiers: Random Forest (RFC), Support Vector Machine (SVM), and Gradient Boosting (GB). The feature numbers used were on five levels, such as 20, 50, 80, 110, and 160. The classification performance was examined in three data distribution scenarios: An equal number of real and fake audio samples, more real audio samples than fake ones, and more fake audio samples than real ones. The results indicated that RFC and GB achieved high

accuracy consistently across all feature sets and cases, especially with the increasing number of features. For instance, in Case 1, RFC and GB achieved 98% to 99% accuracy, while SVM achieved between 94% and 98%. In Case 2, RFC and GB achieved between 97% and 99% performance, while SVM had slightly lower accuracy (94% to 97%). Similarly, in Case 3, RFC and GB achieved accuracy rates of 96% to 97%, while SVM achieved rates of 94% to 96%. The results evidently indicate that feature selection is extremely crucial in model performance enhancement and guaranteeing generalizability, especially in the case of imbalanced datasets. Larger feature sets (i.e., 110 and 160 features) were likely to achieve higher accuracy in the majority of the models, especially for RFC and GB. This suggests that representations with more features enable the classifiers to learn more subtle distinctions between real and fake audio. Meanwhile, it is also important to strike a balance between feature richness and computational efficiency to avoid overfitting or model complexity. Overall, judicious feature selection and tuning are very significant to the robustness and reliability of fake audio detection systems.

## E. Signals Energy Analysis

### 1) Root Mean Square Error (RMSE)

The *RMSE* of an audio signal is calculated using the following formula, as shown in Equation (1)

$$x_{rms} = \sqrt{\left( \frac{1}{n} \sum_{i=1}^{n} (x_i^2) \right)} \qquad (1)$$

Where n is the number of samples and $x_i$ is the i-th sample. For a general signal, *RMSE* is computed as a single value from all the values present in the signal. However, due to the non-stationary nature of audio signals, it is preferable to compute the energy over short intervals to capture the variation over time. Different types of speech may have different energy content, making it an important feature to consider for audio classification.

### 2) Chroma Features

The chroma features represent the frequency spectrum as 12 pitch classes. The entire frequency is divided into 12 bins, denoting the 12 chromas present in the musical octave. The chromagram of a sample audio signal is presented in Figure 4 and Figure 5.

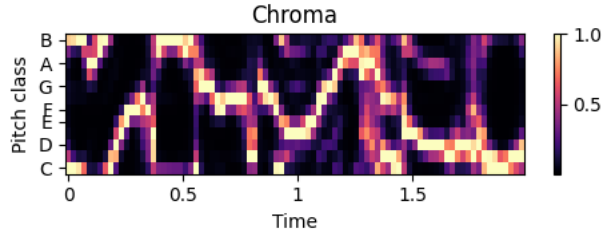


Figure 4. ChromagramoOf a Sample Fake Audio Signal

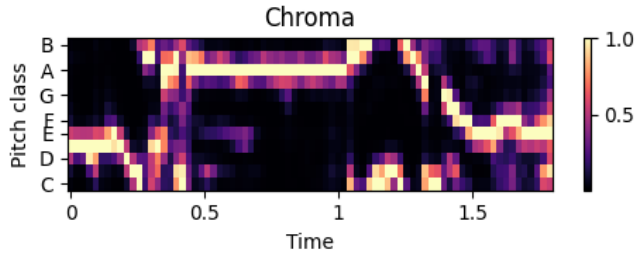


Figure 5. ChromagramoOf a Sample Real Audio Signal

### 3) Spectral Centroid

The spectral centroid is indicative of the nature of the frequency prevalent in the signal. For instance, a higher spectral centroid value indicates a concentration in the high-frequency region of the spectrum. It is computed for the ith audio frame as shown in Equation (2).

$$\mu = \frac{\sum_{k=1}^{k=N} f(k).f(k)}{\sum_{k=1}^{n=N} m(k)} \qquad (2)$$

Where m (k) is the magnitude at the kth frequency bin, and f (k) is the center frequency at the kth frequency bin. The spectral centroid plot of the audio sample is shown in Figure 6 and Figure 7.

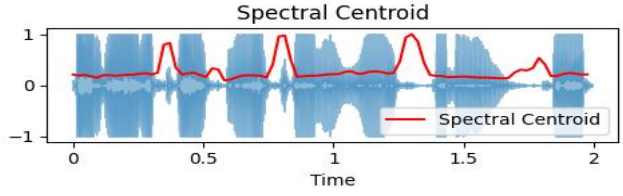


Figure 6. Spectral Centroid Plot of a Sample Fake Audio Signal

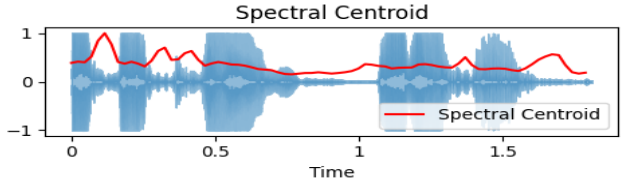


Figure 7. Spectral Centroid Plot of a Sample Real Audio Signal

### 4) Spectral Bandwidth

The range of frequencies in the audio signal spectrum that corresponds to one-half of the peak magnitude of the spectrum is known as the spectral bandwidth of the audio signal. The $P_{\text{th}}$ order spectral bandwidth is given by Equation (3).

$$(\sum_k m(k)(f(k) - \mu)^p))^{(1/p)} \qquad (3)$$

Where m (k) is the magnitude at the kth frequency bin, f (k) is the center frequency at the kth frequency bin, and μ is the Spectral centroid. The spectral bandwidth plot of the fake audio sample is shown in Figure 8 where the spectral bandwidth plot of the real audio sample is shown in Figure 9.

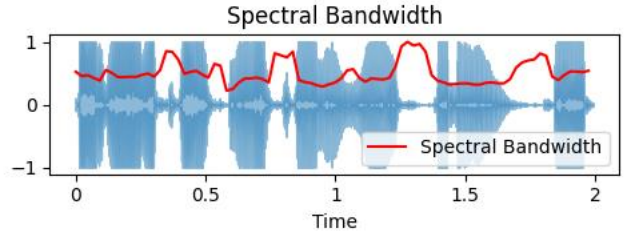


Figure. 8. Spectral Bandwidth Plot of a Sample Fake Audio Signal

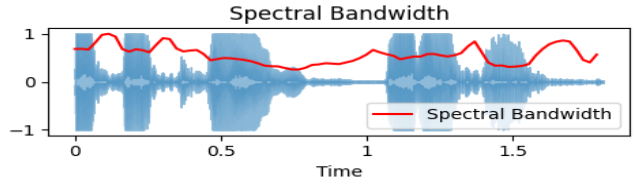


Figure 9. Spectral Bandwidth Plot of a Sample Real Audio Signal

### 5) Spectral Rolloff

The spectral rolloff is the point in the frequency below which 85 percent of the energy present in the spectrum resides. This provides the general shape of the spectra by ignoring the outlier higher frequencies and focusing on the portions of the spectra where most of the energy is present. It is given by Equation (4).

$$\arg \max_{fr \in [1.....N]} \sum_{K=1}^{fr} m(k) 0.85 \sum_{K=1}^{N} m(k) \qquad (4)$$

Where $f_r$ is the rolloff frequency, and *m(k)* is the magnitude at the *k*th frequency bin. The spectral rolloff plot of the sample real audio signal is shown in Figure 10, while the sample fake audio signal is shown in Figure 11**.**

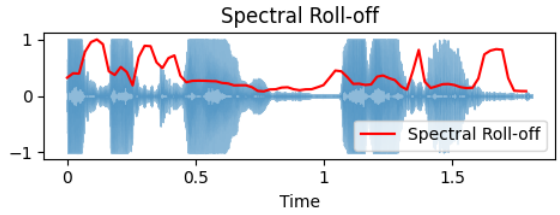


Fig. 10. Spectral Rolloff Plot of a Sample Real Audio Signal

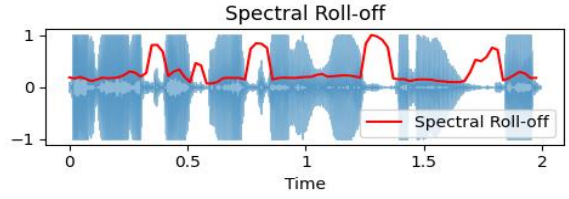


Figure. 11. Spectral Rolloff Plot of a Sample Fake Audio Signal

### 6) Zero Crossing Rate

The rate of sign changes in a signal is referred to as the zero-crossing rate. Lower frequency signals have a lower zero-crossing due to fewer oscillations per second as compared to higher frequency signals. It is given by Equation (5).

$$(1/\ W_L)\ \sum_{n=1}^{WL} |\ \mathrm{sgn}[x(n)] - \mathrm{sgn}[x(n-1)]| \qquad (5)$$

Where x (n) is the audio signal, $W_L$ is the length of the window, and sgn is the signum function. The zero-crossing rate plot of the audio sample is presented in Figure12 and Figure 13**.**

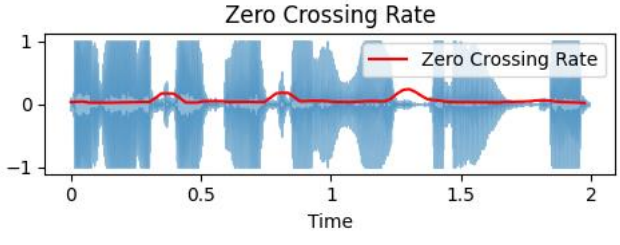


Figure. 12. Zero-Crossing Rate Plot of a Sample Fake Audio Signal

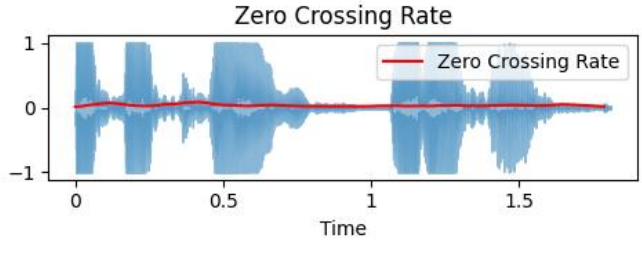

Figure 13. Zero-Crossing Rate Plot of a Sample Real Audio Signal

### 7) Mel Frequency Cepstral Coefficients (MFCCs)

MFCCs reflect the enclosed envelope of the power spectrum that depicts the characteristics of the vocal tract and human voice. Mel-frequency cepstrum, formed by these coefficients, is used to identify periodic components of a time domain signal as peaks in a new domain called the "quefrency" domain (Tusar Kanti Dash, 2021) The MFCCs are obtained by transforming the signal to the frequency domain from the time domain, and then to the quefrency domain from the frequency domain, using a series of mathematical transformations. The process for generating MFCC is similar to that for generating a melspectrogram, with the following addition (Fang Zheng, 2001) : The magnitude of powers from Mel filters is converted to a logarithmic scale. Then, the discrete cosine transform is applied, which outputs cepstral coefficients. Applying the Discrete Cosine Transform (DCT) on the results produced in the preceding stage which results in cepstral coefficients. The MFCC plot of the sample fake audio signal is shown in Figure 14, while the sample of real audio signal is shown in Figure 15.

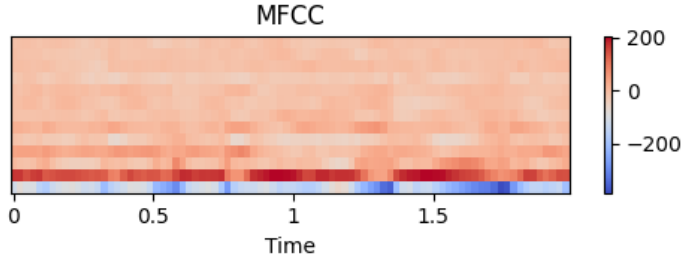


Figure. 14. MFCC Plot of a Sample Fake Audio Signal

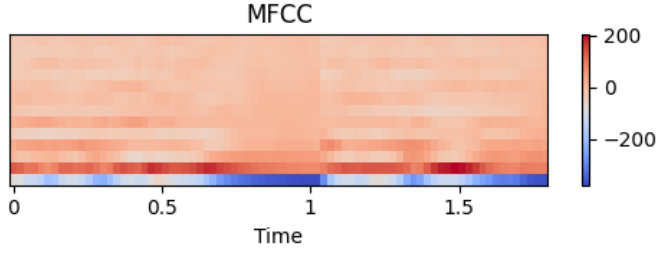


Figure 15. MFCC Plot of a Sample Real Audio Signal

In this study, three well-established machine learning algorithms were used to classify between real and fake audio: Random Forest Classifier (RFC), Support Vector Machine (SVM), *and* Extreme Gradient Boosting (XGBoost). These models were selected due to their proven effectiveness in binary classification tasks, especially in domains where distinguishing subtle variations is critical, such as detecting fake audio (J. Ameer Hamza, 2022) (Suriyaprakash K, 2024)

**a) Random Forest Classifier (RFC):** Random Forest is a decision tree-based algorithm except that it fits many categorizing decision trees on different sub-samples of the dataset and then uses averaging to integrate all the decision trees. It helps in the mitigation of dataset overfitting problems. Random forest is used in calculating the feature importance by adding the gain of each feature and scaling the number of samples passing through the node (J. Ameer Hamza, 2022). It constructs multiple trees during training and aggregates their predictions through majority voting. Its robustness to overfitting, ability to handle high-dimensional data, and effectiveness with unbalanced datasets make it highly suitable for audio classification.

**b) Support Vector Machine (SVM):** SVM is a powerful classification algorithm that seeks the optimal hyperplane to separate classes. It is highly effective in high-dimensional spaces and is known for performing well when class boundaries are clear. However, its performance can be sensitive to noise and data imbalance. Support Vector Machine (SVM) is a supervised learning algorithm that is well known to be among the top discriminative algorithms. SVM for the detection of fake audio is a binary classifier employed to differentiate real and fake audio with the assistance of a decision boundary expressed in terms of a hyperplane. In this case, the attributes such as MFCCs (Mel-Frequency Cepstral Coefficients) and other features are being extracted from original and fake audio signals. Original audio samples are labelled with 0 labels, while the fake ones are labelled with +1 labels. The SVM is applied to this labelled training data to find the optimal separating hyperplane that maximizes the margin (distance) between the two classes (Tianyun Liu, 2021) .SVM is a supervised learning method that relies primarily on two assumptions: 1) Converting data into a high-dimensional space may reduce complex classification issues with complex decision surfaces to more minor problems that may be solved by making it linearly separable, and 2) only training patterns near the decision surface provide the most sensitive details for classification (J. Ameer Hamza, 2022).

**c) Extreme Gradient Boosting (XGBoost):** XGBoost is a scalable and efficient implementation of gradient boosting, where models are built sequentially and optimized to reduce the errors of previous iterations. It includes built-in regularization to prevent overfitting and is suitable for handling large and complex datasets. XGB is a parallel and optimized version of gradient boosting algorithms that combines efficiency and resource management. It implements gradient-boosted decision trees in an iterative model by combining weak base models into a stronger learner. The residual is utilized to refine the loss function and improve the prior prediction at each iteration of the gradient boosting algorithm. It is an approach for gradient boosted decision trees. XGBoost is an algorithm in the class of gradient boosting machines. In boosting algorithms, many weak learners are ensembled sequentially to create a strong learner having low variance and high accuracy. In boosting, the learning of the next predictor is improved to avoid repeating the error caused by any previous predictor. In Random Forest, a model with deeper trees gives good performance, but in XGBoost, shallow trees perform better because of boosting. There are two boosting approaches, Adaptive Boosting and Gradient Boosting. Adaptive boosting puts more weight on misclassified data samples. While gradient boosting identifies misclassified samples as gradients using Gradient Descent to iteratively optimize the loss. XGBoost employs gradient boosting. Using XGBoost will be highly effective for large datasets as it is highly scalable and computationally efficient (Muhammad Umar Farooq, 2025) (Tianqi Chen, 2016). Each algorithm was evaluated using different feature numbers, such as (20, 50, 80, 110, and 160), to explore how the number of features impacts performance. The evaluation was based on four metrics: *accuracy, precision, recall,* and *F1-score,* allowing for a comprehensive performance comparison. The results

confirmed the effectiveness of all three classifiers, with RFC and XGBoost demonstrating slightly better consistency and adaptability across varying data conditions.

## Results and Discussion

This section presents the performance of the machine learning models under different dataset distribution scenarios and feature configurations. The experiments were conducted using three commonly used classifiers: Random Forest Classifier (RFC), Support Vector Machine (SVM), and Gradient Boosting (GB). The evaluation was based on classification accuracy to assess each model's effectiveness in detecting fake audio. The dataset was split using an 80/20 stratified train/test split, ensuring class balance in both subsets.

### a) Model Performance Analysis

The performance of the three models (Random Forest - RFC, Support Vector Machine - SVM, and Gradient Boosting - GB) was visualized across the three dataset cases (equal samples (case 1), more fake samples (case 2), and more real samples (case 3)) using bar charts. These charts illustrate the accuracy of each model relative to the number of features used (20, 50, 80, 110, and 160) as shown in Figure 16.

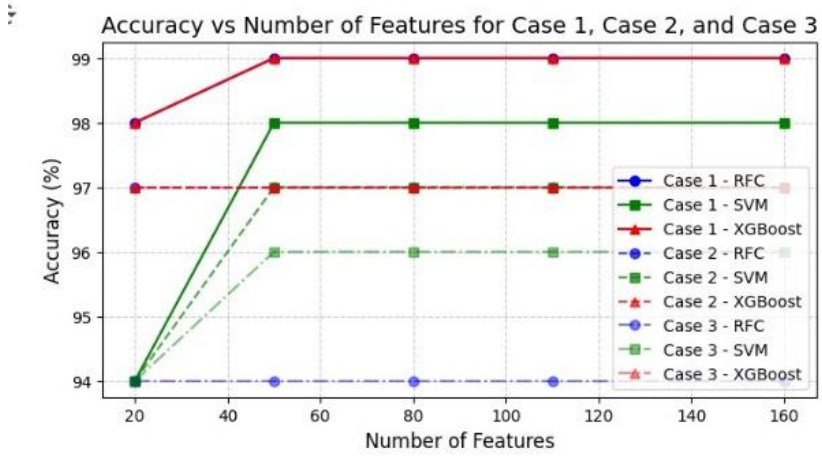


Figure 16.
Accuracy and Number of Features for Different Cases and Classifiers

The results showed that the Random Forest (RFC) algorithm achieved the highest accuracy of 99% in Case 1 (equal samples) when using 50 or more features. In Case 2, its highest accuracy was 97% across all feature counts, and in Case 3, it reached a maximum of 94%, also consistently across all feature sizes, as shown in Figure 17.

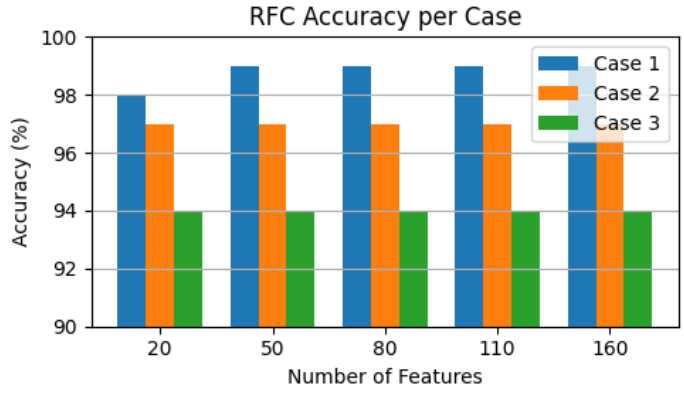


Figure 17. RFC Accuracy per Case

The SVM algorithm recorded its best performance of 98% in Case 1 with 50 or more features, while in Case 2, its highest accuracy was 97%, and in Case 3, it reached up to 96%, as shown in Figure 18.

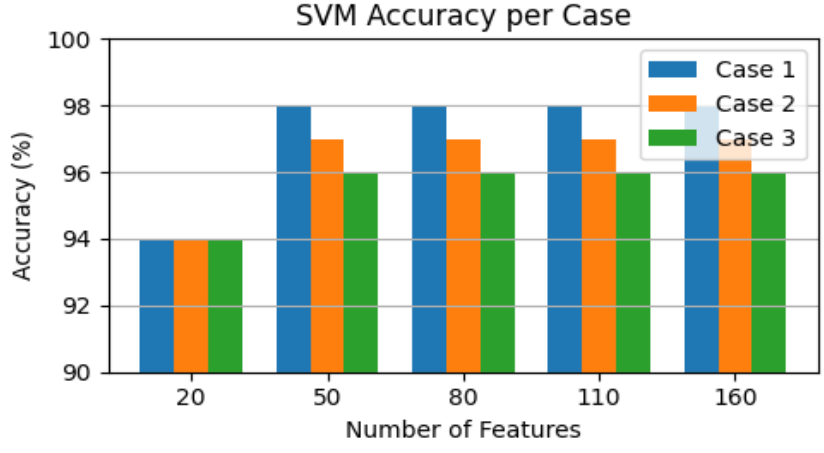


Figure 18. SVM Accuracy per Case

As for Gradient Boosting (GB), it demonstrated relatively stable performance. It achieved its highest accuracy of 99% in Case 1 when using 50 or more features, and maintained 97% in both Case 2 and Case 3 across all feature counts. These findings highlight the significant impact of feature count on model accuracy, particularly in Case 1, and confirm the superior performance of the RFC algorithm in achieving the highest accuracy across different scenarios, as shown in Figure 19.

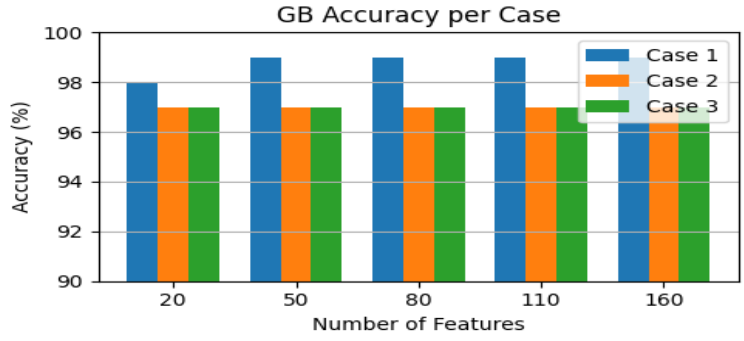


Figure 19. GB Accuracy per Case

**b) Data Scenarios**

**1) First scenario: Balanced dataset (Equal Real and Fake Samples)**

In the balanced scenario, both RFC and GB demonstrated outstanding performance, achieving an accuracy of 99%. These results indicate the effectiveness of ensemble-based methods when the dataset is evenly distributed. SVM, while slightly less accurate, still performed well, achieving an accuracy of approximately 98%, as shown in Table 1.

| Metric | Model | 20 | 50 | 80 | 110 | 160 |
|---|---|---|---|---|---|---|
| **Accuracy** | RFC | 0.98 | 0.99 | 0.99 | 0.99 | 0.99 |
| | SVM | 0.94 | 0.98 | 0.98 | 0.98 | 0.98 |
| | XGB | 0.98 | 0.99 | 0.99 | 0.99 | 0.99 |
| **Precision (0)** | RFC | 0.98 | 0.99 | 0.99 | 0.99 | 0.99 |
| | SVM | 0.94 | 0.99 | 0.99 | 0.98 | 0.98 |
| | XGB | 0.98 | 0.99 | 1.00 | 0.99 | 0.99 |
| **Precision (1)** | RFC | 0.98 | 0.99 | 0.98 | 0.99 | 0.98 |
| | SVM | 0.94 | 0.98 | 0.98 | 0.98 | 0.97 |
| | XGB | 0.98 | 0.99 | 0.99 | 0.99 | 0.98 |

| Metric | Model | 20 | 50 | 80 | 110 | 160 |
|---|---|---|---|---|---|---|
| **Recall (0)** | RFC | 0.98 | 0.99 | 0.98 | 0.99 | 0.99 |
| | SVM | 0.94 | 0.98 | 0.98 | 0.98 | 0.98 |
| | XGB | 0.98 | 0.99 | 0.99 | 0.99 | 0.99 |
| **Recall (1)** | RFC | 0.98 | 0.99 | 0.99 | 0.99 | 0.99 |
| | SVM | 0.94 | 0.99 | 0.99 | 0.98 | 0.97 |
| | XGB | 0.98 | 0.99 | 1.00 | 0.99 | 0.98 |
| **F1-Score (0)** | RFC | 0.98 | 0.99 | 0.99 | 0.99 | 0.99 |
| | SVM | 0.94 | 0.98 | 0.98 | 0.98 | 0.98 |
| | XGB | 0.98 | 0.99 | 0.99 | 0.99 | 0.99 |
| **F1-Score (1)** | RFC | 0.98 | 0.99 | 0.99 | 0.99 | 0.99 |
| | SVM | 0.94 | 0.98 | 0.98 | 0.98 | 0.97 |
| | XGB | 0.98 | 0.99 | 0.99 | 0.99 | 0.98 |

### 2) Scenario 2: Fake-dominant dataset (More Fake than Real Samples)

In the fake-dominant scenario, RFC and GB continued to show strong performance, reaching accuracies of 97%. However, SVM once again showed reduced performance, with an average accuracy of 94%-97%. This reinforces the robustness of ensemble-based models in handling class imbalance, as shown in Table 2.

| Metric | Model | 20 | 50 | 80 | 110 | 160 |
|---|---|---|---|---|---|---|
| **Accuracy** | RFC | 0.97 | 0.97 | 0.97 | 0.97 | 0.97 |
| | SVM | 0.94 | 0.97 | 0.97 | 0.97 | 0.97 |
| | XGB | 0.97 | 0.97 | 0.97 | 0.97 | 0.97 |
| **Precision (0)** | RFC | 0.97 | 0.97 | 0.97 | 0.97 | 0.97 |

| Metric | Model | 20 | 50 | 80 | 110 | 160 |
|---|---|---|---|---|---|---|
| | SVM | 0.95 | 0.98 | 0.98 | 0.98 | 0.98 |
| | XGB | 0.98 | 0.99 | 0.99 | 0.99 | 0.98 |
| **Precision (1)** | RFC | 0.97 | 0.97 | 0.97 | 0.97 | 0.96 |
| | SVM | 0.94 | 0.96 | 0.96 | 0.97 | 0.96 |
| | XGB | 0.96 | 0.97 | 0.97 | 0.97 | 0.96 |
| **Recall (0)** | RFC | 0.93 | 0.93 | 0.93 | 0.93 | 0.92 |
| | SVM | 0.86 | 0.91 | 0.91 | 0.93 | 0.91 |
| | XGB | 0.92 | 0.92 | 0.93 | 0.93 | 0.92 |
| **Recall (1)** | RFC | 0.99 | 0.99 | 0.98 | 0.98 | 0.98 |
| | SVM | 0.98 | 0.99 | 0.99 | 0.99 | 0.99 |
| | XGB | 0.99 | 1.00 | 1.00 | 1.00 | 0.99 |
| **F1-Score (0)** | RFC | 0.95 | 0.95 | 0.95 | 0.95 | 0.94 |
| | SVM | 0.90 | 0.94 | 0.94 | 0.96 | 0.95 |
| | XGB | 0.95 | 0.96 | 0.96 | 0.96 | 0.95 |
| **F1-Score (1)** | RFC | 0.98 | 0.98 | 0.98 | 0.98 | 0.97 |
| | SVM | 0.96 | 0.97 | 0.97 | 0.98 | 0.98 |
| | XGB | 0.98 | 0.98 | 0.98 | 0.98 | 0.98 |

Table 2. Fake-Dominant Dataset (More Fake than Real Samples)

### 3) Scenario 3: Real-Dominant Dataset (More Real than Fake Samples)

When the number of real samples exceeded that of fake ones, a slight drop in classification performance was observed across all models. RFC showed performance, with an accuracy of *94%,* and GB maintained relatively high performance, with an accuracy of *97%,* while SVMs showed performance, with an average accuracy of *94%-96%.* This suggests that imbalanced data can affect the generalization ability of SVM more than ensemble models, as shown in Table 3.

| Metric | Model | 20 | 50 | 80 | 110 | 160 |
|---|---|---|---|---|---|---|
| Accuracy | RFC | 0.94 | 0.94 | 0.94 | 0.94 | 0.94 |
| | SVM | 0.94 | 0.96 | 0.96 | 0.96 | 0.96 |
| | XGB | 0.97 | 0.97 | 0.97 | 0.97 | 0.97 |
| Precision (0) | RFC | 0.95 | 0.95 | 0.95 | 0.95 | 0.95 |
| | SVM | 0.96 | 0.99 | 0.99 | 0.99 | 0.99 |
| | XGB | 0.99 | 1.00 | 1.00 | 1.00 | 1.00 |
| Precision (1) | RFC | 0.93 | 0.93 | 0.93 | 0.93 | 0.93 |
| | SVM | 0.91 | 0.92 | 0.93 | 0.92 | 0.92 |
| | XGB | 0.93 | 0.93 | 0.93 | 0.93 | 0.93 |
| Recall (0) | RFC | 0.94 | 0.94 | 0.94 | 0.94 | 0.94 |
| | SVM | 0.93 | 0.94 | 0.94 | 0.94 | 0.93 |
| | XGB | 0.94 | 0.94 | 0.94 | 0.94 | 0.94 |
| Recall (1) | RFC | 0.94 | 0.93 | 0.93 | 0.93 | 0.93 |
| | SVM | 0.95 | 0.98 | 0.99 | 0.99 | 0.98 |
| | XGB | 0.99 | 1.00 | 0.99 | 0.99 | 1.00 |
| F1-Score (0) | RFC | 0.95 | 0.95 | 0.95 | 0.95 | 0.95 |
| | SVM | 0.94 | 0.96 | 0.97 | 0.96 | 0.96 |
| | XGB | 0.97 | 0.97 | 0.97 | 0.97 | 0.97 |
| F1-Score (1) | RFC | 0.93 | 0.93 | 0.93 | 0.93 | 0.93 |
| | SVM | 0.93 | 0.95 | 0.96 | 0.95 | 0.95 |
| | XGB | 0.96 | 0.96 | 0.96 | 0.96 | 0.96 |

Table 3. Real-Dominant Dataset (More Real Than Fake Samples)

### c) Detailed Description of the Highest Accuracy in Each Case

This part illustrates that the highest accuracy achieved in each of the three study cases was obtained using a specific number of features and a particular classification algorithm or a combination of algorithms, as follows:

#### 1) First case (Equal distribution of real and fake samples)

The highest accuracy of 99% was achieved using 50 features with a combination of Random Forest Classifier (RFC) and Gradient Boosting (GB) algorithms, as shown in Table 4.

#### 2) Second case (More fake audio samples):

An accuracy of 97% was achieved using 20 features, also with the combination of RFC and GB as shown in Table 6.

#### 3) Third case (More Real audio samples):

An accuracy of 97% was obtained using 20 features, but with the Gradient Boosting (GB) algorithm alone, as shown in Table 8.

This analysis highlights the importance of selecting the appropriate number of features and algorithm configuration to maximize detection performance, depending on the data distribution in each case, as shown in Figure 20.

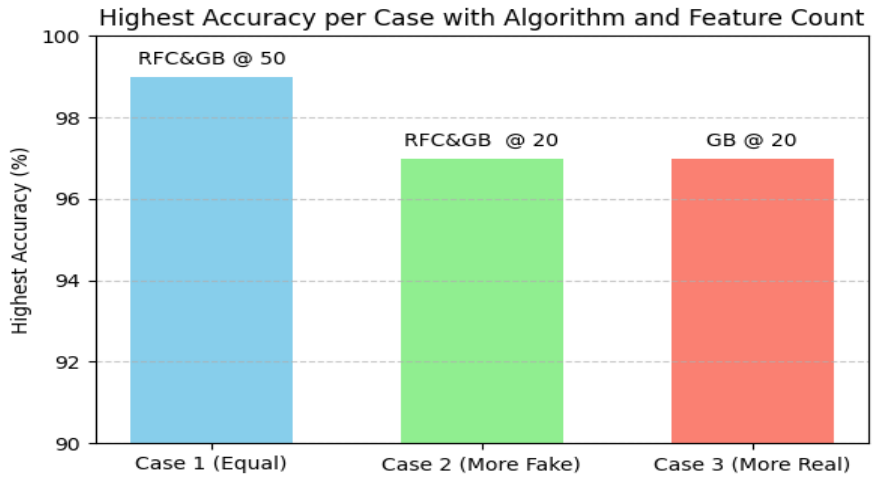


Figure 20. Highest Accuracy per Case with Algorithm and Feature Count

### d) Model Evaluation with K-Fold

A 5-Fold Stratified Cross-Validation is utilized in this study to compare the performance of the models we introduced here, i.e., Random Forest Classifier (RFC), XGBoost, and Support Vector Machine (SVM). The technique involves dividing the dataset into five folds of the same size, maintaining the proportion of classes in each fold (i.e., stratification). The four folds are employed to train the model and are used to test the fifth, and this is repeated five times so that each of the four folds is used once as the test set. The greatest advantage in the use of cross-validation is that it provides an improved and less biased estimate of the model's generalization performance compared to one train-test split.

It also helps in the identification of probable issues such as overfitting or underfitting, and in the stability of the test results that isn't dependent on a specific random split of data. It is especially relevant where the datasets are imbalanced or relatively small in size because it makes maximal use of available data for both training and testing. Besides, single-run evaluation and 5-fold cross-validation were done for all of the models, and the highest results of both approaches were found to be very close, as shown in Tables 5, 6, 7, 8, and 9. This reflects the stability and strength of the performance of the models regardless of the evaluation strategy. Moreover, feature scaling was conducted with the Standard Scalar method before training the models. This preprocessing step ensured features were uniformly scaled to a fixed scale, which is particularly helpful in improving the performance of feature magnitude-sensitive models, especially the SVM.

| **Metric** | **Random Forest (RFC)** | **XGBoost** |
|---|---|---|
| **Testing Accuracy** | 0.99% | 0.99% |
| **Class 0 Precision** | 0.99% | 0.99% |
| **Class 0 Recall** | 0.99% | 0.99% |
| **Class 0 F1-Score** | 0.99% | 0.99% |
| **Class 1 Precision** | 0.99% | 0.99% |
| **Class 1 Recall** | 0.99% | 0.99% |
| **Class 1 F1-Score** | 0.99% | 0.99% |

Table 4. Highest Accuracy Using Single Run (Equal Real and Fake Samples with Feature Number Is 50)

| **Metric** | **Random Forest (RFC)** | **XGBoost** |
|---|---|---|
| **Avg Training Accuracy** | 100.00% ± 0.00 | 100.00% ± 0.00 |
| **Avg Testing Accuracy** | 98.79% ± 0.09 | 99.11% ± 0.07 |
| **Macro Precision** | 98.79% | 99.11% |
| **Macro Recall** | 98.79% | 99.11% |
| **Macro F1-Score** | 98.79% | 99.11% |
| **Class 0 Precision** | 99.03% | 99.36% |
| **Class 0 Recall** | 98.55% | 98.87% |
| **Class 0 F1-Score** | 98.79% | 99.11% |

| | | |
|---|---|---|
| **Class 1 Precision** | 98.55% | 98.87% |
| ***Class 1 Recall*** | 99.03% | 99.36% |
| ***Class 1 F1-Score*** | 98.79% | 99.11% |

Table 5. Highest Accuracy using K-fold (Equal Real and Fake Samples with feature number is 50)

| **Metric** | **Random Forest (RFC)** | **XGBoost** |
|---|---|---|
| **Testing Accuracy** | 0.97% | 0.97% |
| **Class 0 Precision** | 0.97% | 0.98% |
| **Class 0 Recall** | 0.93% | 0.92% |
| **Class 0 F1-Score** | 0.95% | 0.95% |
| **Class 1 Precision** | 0.97% | 0.96% |
| **Class 1 Recall** | 0.99% | 0.99% |
| **Class 1 F1-Score** | 0.98% | 0.98% |

Table 6. Highest Accuracy Using Single Run (More Fake Than Real Samples with Feature Number Is 20)

| ***Metric*** | ***Random Forest (RFC)*** | ***XGBoost*** |
|---|---|---|
| **Avg Training Accuracy** | 98.21% ± 0.03 | 98.03% ± 0.03 |
| **Avg Testing Accuracy** | 96.31% ± 0.20 | 97.11% ± 0.16 |
| **Macro Precision** | 96.24% | 97.54% |
| **Macro Recall** | 95.29% | 95.87% |
| **Macro F1-Score** | 95.74% | 96.64% |
| **Class 0 Precision** | 96.04% | 98.63% |
| **Class 0 Recall** | 92.40% | 92.35% |

| | | |
|---|---|---|
| **Class 0 F1-Score** | 94.18% | 95.38% |
| **Class 1 Precision** | 96.43% | 96.45% |
| **Class 1 Recall** | 98.18% | 99.39% |
| **Class 1 F1-Score** | 97.30% | 97.90% |

Table 7. Highest Accuracy Using K-Fold (More Fake Than Real Samples with Feature Number Is 20)

| **Metric** | **XGBoost** |
|---|---|
| **Testing Accuracy** | 0.97% |
| **Class 0 Precision** | 0.99% |
| **Class 0 Recall** | 0.94% |
| **Class 0 F1-Score** | 0.97% |
| **Class 1 Precision** | 0.93% |
| **Class 1 Recall** | 0.99% |
| **Class 1 F1-Score** | 0.96% |

Table 8. Highest Accuracy using a single run (More Real than Fake Samples with a feature number is 20)

| **Metric** | **XGboost** |
|---|---|
| **Avg Training Accuracy** | 97.08% ± 0.06 |
| **Avg Testing Accuracy** | 95.06% ± 0.26 |
| **Macro Precision** | 94.82% |
| **Macro Recall** | 95.22% |
| **Macro F1-Score** | 94.98% |
| **Class 0 Precision** | 97.13% |

| | |
|---|---|
| **Class 0 Recall** | 94.12% |
| **Class 0 F1-Score** | 95.60% |
| **Class 1 Precision** | 92.50% |
| **Class 1 Recall** | 96.31% |
| **Class 1 F1-Score** | 94.36% |

Table 9. Highest Accuracy Using K-Fold (More Real Than Fake Samples with Feature Number Is 20)

The confusion matrix shown in Figure 21 and Figure 22 summarizes the classification results, revealing a good balance between true positives and low misclassification rates across both classes in the highest results using a single run and 5-fold.

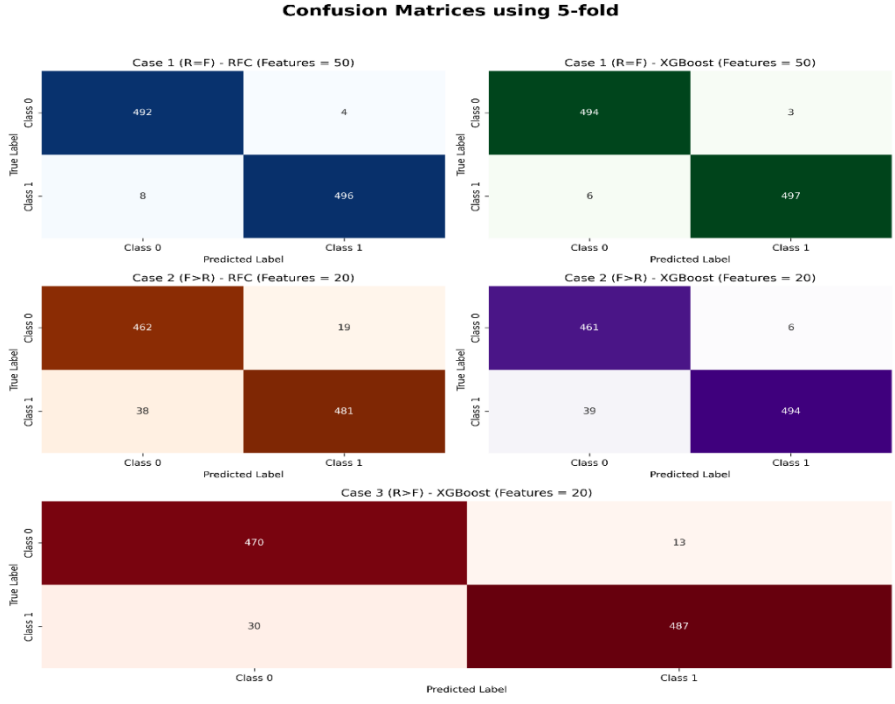


Figure 21. Confusion Matrix Using 5-Fold

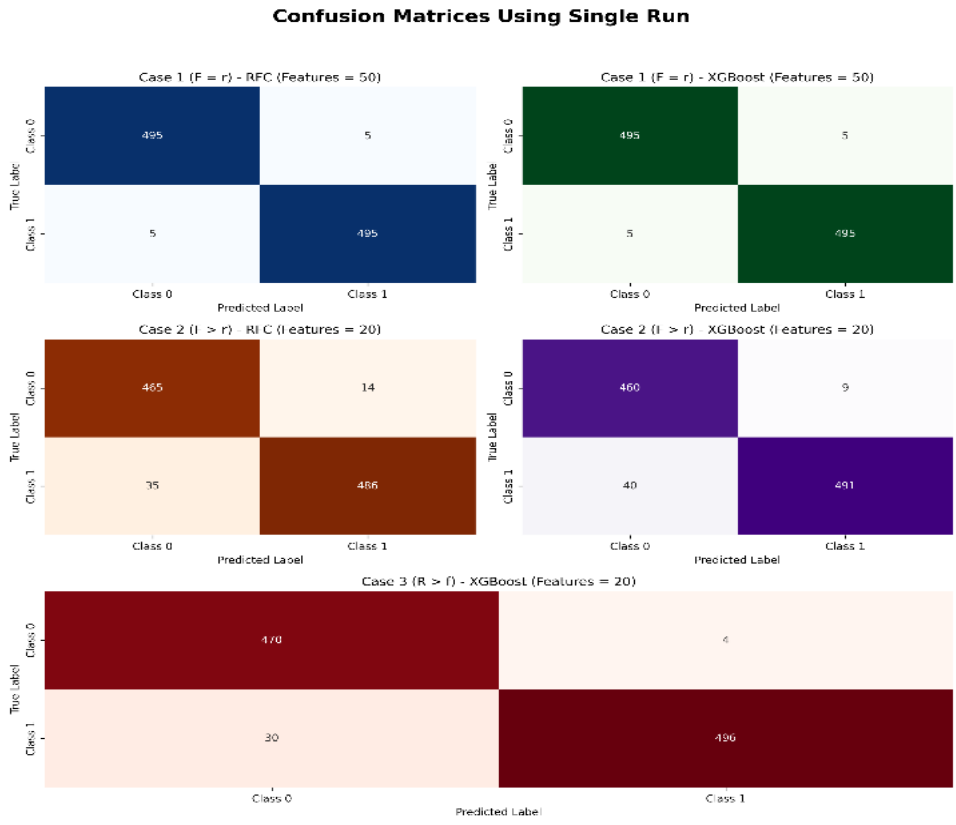


Figure 22. Confusion Matrix Using a Single Run

## Work Limitations

Feature scaling was applied in this study to the dataset prior to model training in order to ensure that all features contributed equally towards the learning process. This was necessary based on the fact that some of the models used, for instance, Support Vector Machines (SVM), are scale-sensitive to the input data. Moreover, k-fold cross-validation was applied to approximate the performance of the models more accurately and reduce the risk of overfitting. However, aside from these preprocessing methods, there are still some limitations. The scaling operation could inadvertently affect the interpretability of the contribution of individual features, especially in using models as a function of feature importance. Further, even though cross-validation increases generalizability, it is computationally costly and fails to eliminate the threat of model bias completely, particularly if the data is infused with underlying imbalances or noise that continues across folds. Hyperparameter tuning was initially considered as a potential step to enhance the model's performance. However, during experimentation, it was observed that the tuning process significantly increased the computation time without yielding a notable improvement in accuracy. Due to the extensive time required for grid or random search methods, and the marginal performance gains obtained from preliminary trials, hyperparameter tuning was not fully explored in this study. This decision represents a limitation, as further optimization of model parameters could potentially improve performance under different configurations or datasets.

## e) Conclusion And Future Works

As the possible threat of audio deepfakes increases in areas such as security, media authenticity, and digital forensics, the need for effective and robust detection mechanisms is more crucial than ever. This study examined how data distribution and feature variation impact the performance of machine learning models in detecting fake audio. Three machine learning models, including Random Forest (RFC), Support Vector Machine (SVM), and Gradient Boosting (GB), were compared across three conditions: balanced datasets, real-dominant datasets, and fake-dominant datasets. The results indicated that RFC and GB performed better than SVM overall, particularly under imbalanced conditions, with RFC and GB achieving up to 99% accuracy on balanced datasets. The study also confirmed that the choice and richness of audio features play an important role in boosting classification performance. Rich spectral and prosodic feature models outperformed models with sparse feature sets. This research highlights the importance of data distribution and feature selection when building fake audio detection systems. In future work, it's planned to extend the evaluation to more challenging and realistic conditions by introducing background noise, transmission distortions, and codec compression. Additionally, exploring deep learning approaches, such as CNNs and Transformer-based models, could further enhance detection performance. Investigating advanced feature selection methods and evaluating cross-dataset and cross-language generalization are also promising directions. Furthermore, incorporating adversarial training could improve the model's robustness against sophisticated attacks, making the system more resilient for real-world deployment. Also, the integration of deep learning models to automatically extract features can be proposed as a one of intended goals, as well as the use of raw audio as input for ease of pre-processing. Hybrid models using machine learning and deep learning techniques can also be anticipated to yield further improvements in detection accuracy and generalization.

## References

Anupama Chadha, V. K. (2021). An overview. In Proceedings of Second International Conference on Computing, Communications, and Cyber-Security.

Dora M. Ballesteros, Y. R.-O. (2021). Deep4SNet: deep learning for fake speech classification.

ELGIBREEN, Z. M. (2023). Detecting Fake Audio of Arabic Speakers Using Self-Supervised Deep Learning.

Fang Zheng, G. Z. (2001). Comparison of Different Implementations of MFCC.

Haoxin Ma, ,. J. (2023). CFAD: A Chinese Dataset for Fake Audio Detection.

Haoxin Ma, J. Y. (2021). Continual Learning for Fake Audio Detection.

J. Ameer Hamza, A. R. (2022). Deepfake Audio Detection via MFCC Features Using Machine Learning.

Janavi Khochare, C. J. (2021). A Deep Learning Framework for Audio Deepfake Detection.

Jonathan Shen, R. P.-R. (2017). Natural TTS Synthesis by Conditioning WaveNet on Mel Spectrogram Predictions.

Lyu, S. (2020). DeepFake Detection: Current Challenges and Next Steps.

Muhammad Umar Farooq, A. J. (2025). A Lightweight and Interpretable Deepfakes.

Mvelo Mcubaa, A. S. (2023). The Effect of Deep Learning Methods on Deepfake Audio .

Nicholas Diakopoulos, D. J. (2020). Anticipating and Addressing the Ethical Implications of Deepfakes in the Context of Elections.

Ricardo Reimao, V. T. (2019). FoR: A Dataset for Synthetic Speech Detection.

Stupp, C. (2019). Fraudsters Used AI to Mimic CEO's Voice in Unusual Cybercrime Case.

Supasorn Suwajanakorn, S. M.-S. (2017). Synthesizing Obama: learning lip sync from audio.

Suriyaprakash K, S. S. (2024). Advanced Deep Fake Detection: Leveraging Xgboost and SVM.

Swadhin Pradhan, W. S. (2019). Combating Replay Attacks Against Voice Assistants.
Tianqi Chen, C. G. (2016). XGBoost: A Scalable Tree Boosting System.
Tianxiang Chen, A. K. (2020). Generalization of Audio Deepfake Detection.
Tianyun Liu, D. Y. (2021). Identification of Fake Stereo Audio Using SVM and CNN.
Tusar Kanti Dash, S. M. (2021). Detection of COVID-19 from speech signal using bio-inspired based cepstral features.
Wei Ping, K. P. (2017). Deep Voice 3: Scaling Text-to-Speech with Convolutional Sequence Learning.
Yohanna Rodríguez-Ortega, D. M. (2020). A Machine Learning Model to Detect Fake Voice.
Zahra Khanjani, G. W. (2021). How Deep Are the Fakes? Focusing on Audio Deepfake: A Survey.
Zaynab Almutairi, H. E. (2022). A Review of Modern Audio Deepfake Detection Methods: Challenges and Future Directions.