Computers, Materials & Continua

Tech Science Press

https://doi.org/10.32604/cmc.2026.080593

ARTICLE

Fine-Tune Transfer Learning Model for Deepfake Audio Detection Using Hybrid Features and Data Augmentation

Rashid Jahangir[1,*], **Nazik Alturki**[2] **and Muhammad Zubair Khan**[1]

[1]Department of Computer Science, COMSATS University Islamabad, Vehari Campus, Vehari, Pakistan
[2]Department of Information Systems, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia

*Corresponding Author: Rashid Jahangir. Email: rashidjahangir@cuivehari.edu.pk

Received: 12 February 2026; Accepted: 12 May 2026; Published: 23 July 2026

ABSTRACT: Deepfake audio created with sophisticated speech synthesis and voice cloning technologies is a threat to the credibility of digital communication. Its realism has raised serious concerns in different applications such as digital forensics, cybersecurity, media authentication and voice-based security systems. However, deepfake audio detection still remains difficult. Synthetic speech tends to have subtle artifacts that can mimic the natural vocal pattern very closely. Variations in speakers, recording conditions and background noise make the task more complex. In addition, dataset imbalance and low diversity in training samples could lead to low robustness in the model. To overcome these limitations, the present study aims to propose a framework of transfer learning-based methods based on a combination of fine-tuned pre-trained models, as well as systematic data augmentation. Augmentation methods are introduced to increase the variability and mimic real acoustic conditions. This approach supports the learning of more stable and generalizable representations for both genuine and manipulated speech. The framework employs three DL models: ResNet50 to capture global spectro-temporal structures, VGGish to extract mid-level semantic audio embeddings and YAMNet to identify fine-grained temporal irregularities associated with synthetic speech artifacts. Features from these models are fused through concatenation to construct a unified hybrid feature space. A feature selection stage then reduces redundancy before classification using a lightweight model. Experimental results demonstrate the superiority of the proposed hybrid approach and achieved an accuracy of 99.7%. This performance significantly outperformed individual baseline models and achieved strong generalization across diverse acoustic conditions.

KEYWORDS: Deepfake audio detection; transfer learning; hybrid feature fusion; data augmentation; audio forensics; synthetic speech detection

1 Introduction

The human voice plays a central role in communication, identity, and trust. In recent years, deepfake audio technologies have created a serious threat to the authenticity of spoken content. Deepfake voice refers to artificially generated or manipulated audio that imitates the speech of a real person using generative machine learning techniques [1,2]. These systems can reconstruct a person's tone, prosody, and speaking style from only a few seconds of recorded audio, making synthetic speech highly convincing and difficult to distinguish from genuine recordings [1,3]. While these technologies support useful applications in entertainment, accessibility, and human–computer interaction, their misuse raises major ethical and security concerns. Criminals and malicious actors use fake audio to commit fraud, spread misinformation, influence public opinion, and fabricate evidence in sensitive domains such as politics and law enforcement [2,4].

Copyright © 2026 The Authors. Published by Tech Science Press. This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Detecting deepfake audio remains difficult because speech signals are complex and dynamic. Unlike images or videos, speech cannot be visually inspected frame by frame. Speech varies across speakers, languages, recording devices, and environments. Subtle changes in pitch, timing, and frequency often escape human hearing yet influence automated analysis. These properties make deepfake audio detection a challenging research problem [5].

Early detection approaches relied on statistical models and hand-crafted features such as Mel-Frequency Cepstral Coefficients (MFCCs). Although these techniques offered initial success, they struggled to keep pace with modern deepfake generators that reproduce fine-grained speech characteristics [6]. Researchers later adopted deep learning models such as Convolutional Neural Networks (CNNs) and attention-based architectures that learn representations directly from spectrograms or raw audio signals. These models achieve higher detection accuracy by capturing hierarchical spectral and temporal patterns. However, they require large labeled datasets, significant computational resources, and often fail to generalize to unseen types of synthetic audio [4,7].

Transfer learning addresses many of these limitations. Instead of training a model from scratch, researchers adapt a pre-trained network to the deepfake detection task by fine-tuning learned feature representations. This strategy reduces training cost and lowers data requirements. Pre-trained models such as ResNet50 and other deep convolutional architectures have demonstrated strong feature extraction capability when applied to deepfake detection [8]. Combining features from multiple architectures further improves robustness because each model captures different aspects of speech signals [5,7]. Dataset imbalance remains another major challenge. Most public datasets contain more real audio samples than fake ones, biasing model training and reducing detection performance. Data augmentation techniques such as pitch shifting, time stretching, noise addition, and spectrogram modification increase the diversity of synthetic samples and create a more balanced training set. Balanced data allows models to learn representative patterns from both real and synthetic speech [3,6].

This research proposes a transfer learning framework for deepfake voice detection. The framework fine-tunes three pre-trained models—ResNet50, VGGish [5], and YAMNet [7]—to extract complementary features from audio spectrograms. It then combines the most informative features into a unified hybrid feature space [9] and trains a lightweight classifier on top. The study applies data augmentation before feature extraction to handle class imbalance and improve generalization. This design seeks to achieve high detection accuracy while maintaining computational efficiency suitable for real-world applications such as digital forensics, security systems, and media verification. This work extends beyond technical performance. Deepfake audio erodes trust in digital communication and threatens social, political, and economic stability. Effective detection systems must therefore deliver accuracy, efficiency, and resilience to new attack methods. This study demonstrates that fine-tuned transfer learning combined with hybrid feature extraction and data augmentation provides a practical and scalable solution for detecting deepfake voices and protecting individuals and institutions. The novelty of the proposed framework lies in the construction of a multi-scale hierarchical feature space. While conventional approaches often focus on a single acoustic dimension, framework uniquely synchronized three distinct levels of audio abstraction. The main contributions of this research are summarized as follows:

- This study proposed a novel hybrid framework that integrated global spectro-temporal structures, mid-level semantic audio embeddings, and fine-grained temporal irregularities to capture a comprehensive set of synthetic artifacts.
- A systematic augmentation strategy was implemented to ensure the model remains resilient to varied recording conditions.

- The ANOVA F-tests were utilized to reduce high-dimensional embeddings into a compact, high discriminative feature space. This allowed the framework to achieve state-of-the-art performance with significantly reduced computational complexity.

The remainder of this study is organized as follows. Section 2 reviews related work on deep learning approaches for audio forgery detection. Section 3 describes the feature extraction networks, dimensionality reduction techniques, and classification methods. Section 4 presents the experimental results and discussion. Finally, Section 5 concludes the study and outlines directions for future research.

2 Literature Review

Audio deepfake generation methods were broadly divided into two categories: text-to-speech synthesis and voice conversion. Text-to-speech systems generated speech directly from textual input. Early systems relied on audio concatenation and produced speech that sounded artificial and discontinuous. Later neural architectures such as WaveNet [10] and Tacotron 2 [11] greatly improved speech realism by learning from large speech datasets. Subsequent models, including autoencoders, autoregressive networks, and Parallel WaveNet [12], increased both efficiency and audio fidelity. Systems such as Deep Voice [13] and Google's Tacotron adopted sequence-to-sequence learning with attention mechanisms. These approaches produced high-quality synthetic speech but required considerable computational resources. Voice conversion followed a different strategy. These systems modified existing speech so that it resembled a target speaker while preserving the original linguistic content. Researchers commonly used methods such as Gaussian Mixture Models [14] and StarGAN-VC [15]. Many voice conversion techniques depended on paired datasets in which source and target speakers uttered identical sentences, allowing accurate spectral mapping. Later studies demonstrated promising results in cross-lingual voice conversion. Although these methods supported useful applications such as dubbing and voice assistants, researchers raised serious concerns about their misuse for unauthorized voice imitation.

2.1 Handcrafted Acoustic Feature-Based Methods

Early deepfake audio detection methods relied heavily on handcrafted acoustic features. Researchers extracted features such as Mel-Frequency Cepstral Coefficients, Constant-Q Transform representations, and spectral descriptors. They classified these features using Logistic Regression, Gaussian Mixture Models, and Support Vector Machines. The Half-Truth Audio Detection dataset [16] exposed a key limitation of these approaches. Detection accuracy dropped sharply when attackers manipulated only portions of an audio signal. Experiments with GMM-based systems and Light CNN models [17] showed that partial manipulation proved harder to detect than fully genuine or fully fake audio. These methods also struggled with noise, varied recording conditions, and unseen attack types. Their reliance on static, hand-designed features limited scalability in real-world scenarios.

2.2 Deep Learning and Spectrogram-Based Architectures

Deep learning later shifted detection toward automatic feature learning. Researchers applied models such as Convolutional Neural Networks, Recurrent Neural Networks, and Convolutional Recurrent Neural Networks to learn representations directly from spectrograms or raw waveforms. This transition reduced dependence on manual feature engineering. Studies using the ASVspoof2019 [3] dataset showed that CNN–RNN hybrid models effectively captured both spectral and temporal speech characteristics. Several studies combined handcrafted features with deep embeddings derived from self-supervised models such as wav2vec [18] and HuBERT [19], which improved cross-dataset generalization. New datasets such as

FakeSound [20] expanded research beyond speech to manipulated environmental sounds. End-to-end architectures, including Res-TSSDNet and modified Inception-based networks, achieved strong performance on short audio segments but required high computational power.

2.3 Transfer Learning and Self-Supervised Models

Hybrid detection models emerged as an effective direction. Researchers integrated CNNs, RNNs, and optimization techniques to improve performance across datasets. Some studies employed Particle Swarm Optimization to refine hybrid model parameters. Self-supervised frameworks such as PASE+ extracted contextual acoustic features and supported scalable deployment. Ensemble learning further increased robustness. Systems that combined spectrogram-based CNNs, transfer learning models, and pre-trained embeddings such as Whisper [21] and SpeechBrain consistently outperformed individual models and achieved very low error rates on ASVspoof2019. Transfer learning played a major role in modern deepfake detection. Researchers adapted pre-trained models to spoof detection tasks instead of training networks from scratch. This approach reduced data requirements and training costs. Methods based on wav2vec 2.0 combined with Variational Information Bottleneck techniques improved both robustness and efficiency. Feature-mismatch strategies such as the SLIM framework [22] exploited inconsistencies between speaking style and linguistic content. Data augmentation supported these approaches. Techniques including pitch shifting, time stretching, and noise injection balanced datasets and increased variability, which helped address persistent class imbalance [6]. Despite substantial progress, several challenges remained. Classical machine learning methods failed to adapt to rapidly evolving attack techniques. Deep learning models demanded high computational resources and remained sensitive to dataset bias and adversarial manipulation. Hybrid deepfakes, in which attackers altered only segments of audio, continued to challenge detection systems. Real-time deployment and generalization to unseen attack methods also remained unresolved. Researchers therefore explored lightweight architectures, adversarial training methods, and multimodal fusion strategies.

2.4 Research Gaps

The reviewed literature revealed several research gaps which this study aims to address:

- Most models focused on either global spectral features or local temporal artifacts. However, the combination of both global and local temporal features is not explored in this field.
- High-performing deep learning systems require extensive computation and large datasets which makes them difficult for real-time deployment.
- There is a persistent need for robust augmentation strategies to handle the scarcity of high-quality manipulated samples.

This study addressed these gaps by proposing a fine-tuned transfer learning framework that integrated hybrid feature extraction with systematic data augmentation. The framework combined features from ResNet50, VGGish, and YAMNet into a unified representation while augmentation balanced the dataset. The framework targeted accurate, efficient, and generalizable detection of audio deepfakes.

3 Proposed Methodology

The proposed methodology adopted for deepfake audio detection is shown in Fig. 1. In first step, the preprocessing and augmentation was performed on DeepfakeDetection-Audio dataset to improve data quality and maintain balance between real and fake samples. The study applied augmentation techniques such as pitch shifting, noise injection, time stretching, and SpecAugment to increase variability in the

training data and reduce class bias. In second step, the audio signals were converted into Mel-spectrograms. This representation captured both temporal and frequency-related speech characteristics and provided a structured format suitable for deep learning models. Afterwards, these spectrograms were used as input to four pre-trained networks: ResNet50, VGGish, YAMNet and a hybrid model. The hybrid model fused the outputs of these networks into a unified hybrid feature space. This fusion combined complementary information, including spectral structure, temporal variation, and contextual acoustic cues. Finally, the hybrid features were given to a lightweight classifier. The classifier achieved high detection accuracy while maintaining low computational cost. The detail of each phase is presented in subsequent subsections.

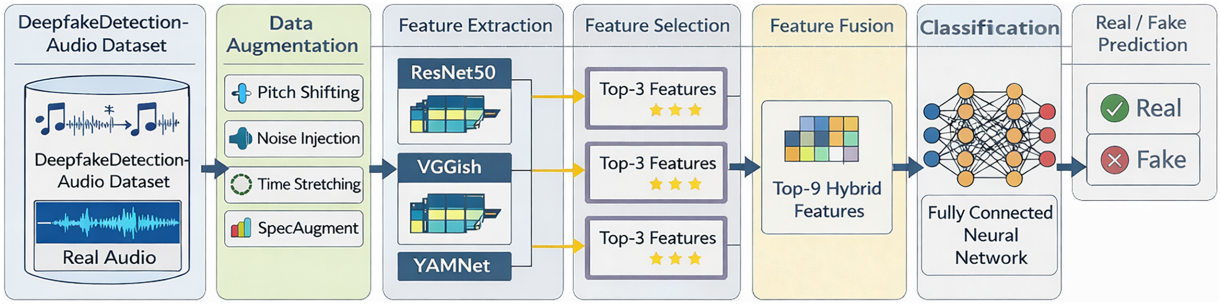

Figure 1: Workflow of proposed research methodology.

3.1 Dataset

This used the "In-the-Wild" corpus [23], a publicly available (https://www.kaggle.com/datasets/abdallamohamed312/in-the-wild-audio-deepfake) benchmark to evaluate synthetic speech detection models. The dataset contained 31,800 audio samples, divided between authentic human speech and artificially generated deepfake voices. Specifically, this dataset offered a diverse set of audios from 58 celebrities, politicians, and other public figures. These samples were collected from social networks and video streaming platforms to ensure real-world diversity. In total, the corpus comprised of 38 h of audio, partitioned into 20.8 h of real and 17.2 h of fake audio. These characteristics made it a reliable data for training and evaluation of proposed deepfake detection models.

3.2 Data Augmentation

Human speech naturally varied in pitch, speaking rate, intonation, and recording quality because speakers, environments, and recording devices differed. By simulating this variability, augmentation allowed the model to learn from a wider range of acoustic conditions and improved its ability to distinguish genuine speech from manipulated audio. When datasets were limited or imbalanced, models tended to favor the majority class. Augmentation addressed this issue by introducing controlled diversity and expanding the dataset without requiring manual annotation. This synthetic variability enhanced the model's capacity to identify subtle artifacts introduced by speech synthesis and voice conversion systems. The commonly applied techniques are Gaussian noise injection, pitch shifting, and time stretching in deepfake detection studies. Gaussian noise injection replicated real-world acoustic conditions such as background conversations, environmental sounds, and electronic interference. This method controlled the noise amplitude using the hyperparameter σ. This parameter was tuned carefully because excessive noise masked important vocal characteristics, while insufficient noise failed to introduce meaningful variation. In implementation, $\sigma = 0.005$ was used to ensure a realistic noise-to-signal ratio. Mathematically, it is presented in Eq. (1):

$$x_{\text{aug}} = x + \mathcal{N}(0, \sigma^2) \tag{1}$$

where x represents the original audio and $\mathcal{N}(0, \sigma^2)$ represents normally distributed random noise. Secondly, pitch shift modified the fundamental frequency of a speech signal without duration alteration to create natural variations in tone. This process helped the model to learn features that remained invariant to speaker identity and pitch range. This technique is used to simulate differences in vocal characteristics across speakers. Random shifts were applied within a range of ±2 semitones. The transformation is given in Eq. (2):

$$x_{\text{aug}}(t) = x(t) \cdot e^{j2\pi\Delta f t} \tag{2}$$

In this equation, Δf represents the applied frequency shift. Moderate pitch adjustments increased variability and improved model robustness, whereas extreme shifts produced unnatural speech patterns and degraded performance. Finally, time stretching modified the speed of speech and introduced variations in rhythm and tempo. This technique reflected natural differences in speaking styles, from slow and deliberate speech to faster utterances. This study utilized time-scaling factors $\alpha \in \{0.9, 0.95, 1.05, 1.1\}$. Mathematically, the transformation is given in Eq. (3).

$$x_{\text{aug}}(t) = x(\alpha t) \tag{3}$$

where α is the time-scaling factor ($\alpha < 1$ slows down the audio, $\alpha > 1$ speeds it up). To further enhance robustness, time shifting was implemented using random circular shifts between 100 and 300 ms, and volume scaling using a random gain factor $g \in [1.1, 1.5]$. Additionally, convolutional reverberation was applied using a synthetic impulse response of 0.5 s to simulate various indoor acoustic environments. Following augmentation, a five-step normalization protocol was applied, involving silence trimming, peak normalization to 0 dBFS, RMS normalization, and soft-clipping via a hyperbolic tangent (tanh) function to ensure numerical consistency across the balanced corpus.

3.3 Feature Extraction

An important phase of this research involved the extraction of discriminative features that separated real speech from deepfake audio. Traditional handcrafted features such as MFCCs or LPCs captured only shallow spectral information. In contrast, deep learning models learned hierarchical representations that encoded fine-grained acoustic variations along with higher-level semantic patterns. To utilize knowledge embedded in established architectures, the study adopted a transfer learning strategy and employed three pre-trained networks: ResNet50, VGGish, and YAMNet. These models were originally trained on large-scale datasets such as ImageNet and AudioSet, which enabled them to extract general and informative features. However, directly using pre-trained embeddings did not adequately support deepfake detection because those features were not optimized to capture synthesis artifacts. Therefore, the study fine-tuned each network to adapt its learned representations to the target task. This fine-tuning played a crucial role because it connected general acoustic representations with task-specific cues required to identify manipulated speech.

3.3.1 Feature Extraction Using ResNet50

The Residual Network with 50 layers (ResNet50) was introduced in 2015 as a major advancement in training deep CNNs. Earlier CNNs suffered from vanishing gradients as network depth increased, which caused training performance to degrade. ResNet50 addressed this limitation through residual learning, where skip connections allowed the input of a block to bypass intermediate layers and be added directly to the output. This structure reduced gradient decay and enabled stable training of deeper networks. This study transformed raw waveforms into Mel-spectrograms derived from speech signals as shown in Fig. 2 and fed as input to ResNet50. A Mel-spectrogram represented audio as a two-dimensional time–frequency image,

where the horizontal axis denoted time, the vertical axis represented frequency on the perceptual Mel scale, and color intensity reflected signal energy. The model used computer vision techniques to analyze speech as structured time–frequency patterns by converting audio into this image-like form. Differences between genuine and synthetic speech often appeared as visual artifacts in spectrogram which made CNN-based models effective for this task.

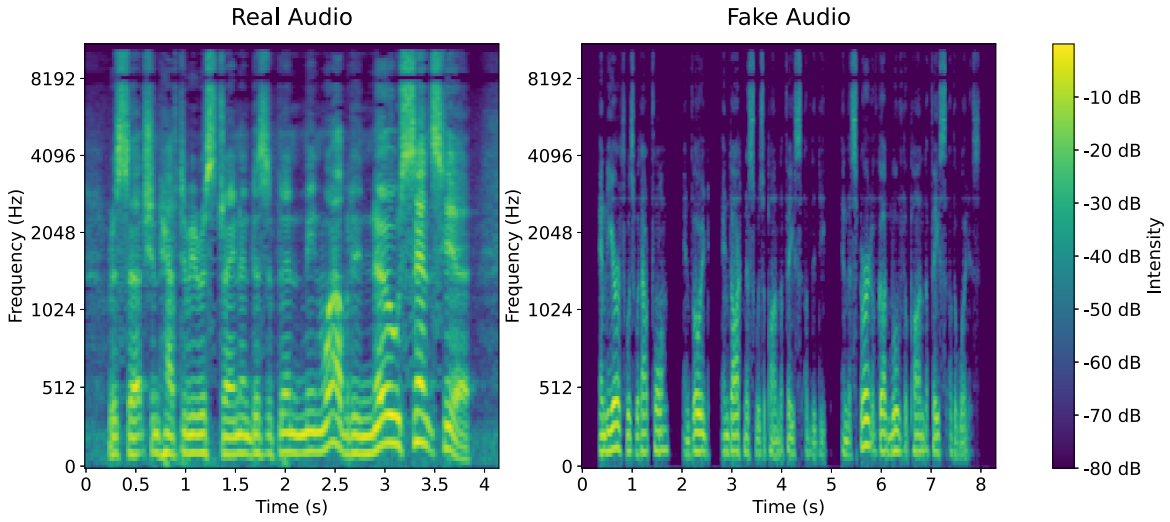

Figure 2: Comparative Log-Mel spectrogram analysis of authentic and synthetic audio samples.

The fundamental residual block in ResNet50 is given in Eq. (4).

$$H(x) = F(x, W) + x, \tag{4}$$

where x is the input vector, $F(x, W)$ is the residual mapping parameterized by convolutional filters W, and $H(x)$ is the block output. The convolutional operations within each block are defined as:

$$y_{i,j,k} = \sum_{m} \sum_{n} \sum_{c} x_{i+m,j+n,c} \cdot w_{m,n,c,k} + b_k, \tag{5}$$

where x is the input tensor, w represents the learnable filter weights, b_k is the bias term, and $y_{i,j,k}$ is the resultant activation. These activations are normalized using batch normalization (BN), which standardizes input distributions:

$$\hat{x} = \frac{x - \mu}{\sqrt{\sigma^2 + \epsilon}}, \tag{6}$$

where μ and σ^2 are batch values. This study removed the final classification layers designed for classification from ResNet50. Instead, the last convolutional feature maps were passed through a Global Average Pooling (GAP) layer given in Eq. (7).

$$f_{\text{ResNet}} = \frac{1}{N} \sum_{i=1}^{N} h_i, \tag{7}$$

where h_i represents the activations from the final convolutional stage and N is the number of elements. This operation compressed the spatial feature maps into a single 2048-dimensional embedding vector, which is then normalized using Eq. (8).

$$\hat{f}_{\text{ResNet}} = \frac{f_{\text{ResNet}}}{\|f_{\text{ResNet}}\|_2}. \tag{8}$$

The lower convolutional layers were froze because they captured general low-level patterns. The deeper convolutional blocks were unfrozen and retrained on the deepfake dataset. This selective fine-tuning strategy allowed the network to adjust to the distribution of real and synthetic spectrograms. Moreover, it also prevented catastrophic forgetting of general audio structures. The process involved retraining only the final one or two residual stages, where the network formed high-level task-specific representations. These modifications produced a useful impact. After fine-tuning, the embeddings captured discontinuous harmonic patterns, unnatural prosody transitions, and inconsistencies in background noise.

3.3.2 Feature Extraction Using VGGish

VGGish was employed as an audio feature extraction model derived from the VGG16 architecture. Instead of large convolution kernels, VGG used small 3×3 filters stacked sequentially. This architecture achieved a large effective receptive field by depth expansion through these smaller filters. Moreover, it also controlled parameter growth and supported better generalization. Google later adapted this concept into VGGish and trained it on millions of labeled audio clips across thousands of sound categories. The VGGish model first resampled the audio waveform to 16 kHz and divided it into overlapping frames. It then converted each frame into a log Mel-spectrogram of size 96×64. These spectrogram served as image-like inputs to the network. The early convolutional layers extracted local spectral structures, while deeper layers learned higher-level abstractions including pitch trajectories, energy contours, and broader acoustic events. Mathematically, the convolution operation in VGGish can be expressed as:

$$y_l = \sigma\left(W_l * x_{l-1} + b_l\right), \tag{9}$$

where x_{l-1} is the input feature map from the previous layer, W_l and b_l are the convolutional kernel and bias, and σ is the ReLU activation function. Each convolutional block in VGGish consists of two such convolution layers followed by max pooling.

$$y_{i,j} = \max_{m,n}\left(x_{i+m,j+n}\right). \tag{10}$$

The Eq. (10) is used to reduce spatial resolution and retain salient spectral features. After four convolutional blocks, the feature maps are flattened and passed through three fully connected layers to produce a embedding vector using Eq. (11).

$$f_{\text{VGGish}} \in \mathbb{R}^{128}. \tag{11}$$

These embeddings were widely adopted in audio research as representations of acoustic events. However, they did not specifically target artifacts produced by synthetic speech systems. As a result, their direct use did not provide sufficient discrimination for deepfake detection. To adapt VGGish to the deepfake voice detection task, the study fine-tuned the model by adding task-specific dense layers. The framework passed the original 128-dimensional embeddings through fully connected layers with 512 and 256 neurons. Dropout layers with rates of 0.3 and 0.2 were inserted between these layers to reduce overfitting. The final classification stage used a single neuron with sigmoid activation to perform binary classification between real and fake speech. The fine-tuned mapping is written as:

$$f_{\text{VGGish-tuned}} = \sigma\left(W_2\phi(W_1 f_{\text{VGGish}} + b_1) + b_2\right), \tag{12}$$

where ϕ is the ReLU function and σ is the sigmoid activation. The fine-tuned VGGish contributed semantic-level representations that complemented the global structural features extracted by ResNet50. VGGish effectively encoded pitch, energy dynamics, and background consistency where synthetic speech often showed weaknesses. However, the relatively low dimensionality of its 128-dimensional embeddings limited its standalone discriminative capacity.

3.3.3 Feature Extraction Using YAMNet

YAMNet is a lightweight deep learning model for audio event classification. Unlike heavier models such as VGG, YAMNet was designed for efficiency and portability which makes it suitable for real-time applications and deployment in resource-constrained environments. In YAMNet, the model first resampled the audio input to 16 kHz and converted it into log Mel-spectrograms with 64 Mel-frequency bins. The network processed these spectrograms through a sequence of depthwise separable convolution layers. Each layer divided a conventional convolution into two steps. The first step applied a depthwise convolution independently to each input channel, and the second step applied a pointwise convolution to combine information across channels (1×1 kernel) that recombines channels. Mathematically, the depthwise convolution is defined as:

$$y_{i,j}^{(k)} = \sum_m \sum_n x_{i+m,j+n}^{(k)} \cdot w_{m,n}^{(k)}, \tag{13}$$

where each input channel k is filtered independently. The subsequent pointwise convolution is expressed as:

$$z_{i,j,c} = \sum_k y_{i,j}^{(k)} \cdot w_{k,c}. \tag{14}$$

In its pre-trained configuration, YAMNet produced 1024-dimensional embeddings that represented general acoustic events. These embeddings were optimized for broad sound classification rather than artifacts introduced by synthetic speech generation. To adapt the model to the deepfake detection task, the study fine-tuned the network and added task-specific fully connected layers. It appended dense layers with 512 and 256 neurons, each using ReLU activation, and placed dropout layers with rates of 0.3 and 0.2 between them to reduce overfitting. The framework then used the output of the 512-dimensional dense layer as the refined embedding for classification. The transformation can be represented as:

$$f_{\text{YAMNet-tuned}} = \phi\left(W_2\phi\left(W_1 f_{\text{YAMNet}} + b_1\right) + b_2\right), \tag{15}$$

where $f_{\text{YAMNet}} \in \mathbb{R}^{1024}$ is the pre-trained embedding, ϕ denotes ReLU, and the tuned embedding $f_{\text{YAMNet-tuned}} \in \mathbb{R}^{512}$. Fine-tuning proved essential because deepfake audio manipulations often appeared at fine temporal scales and produced subtle irregularities in pitch transitions, phase continuity, and short-term spectral energy. The lightweight structure of YAMNet and its adapted embeddings enabled the model to capture these local anomalies effectively. YAMNet showed strong sensitivity to micro-level distortions that the global representations of ResNet50 and the semantic embeddings of VGGish sometimes overlooked.

3.4 Feature Selection and Fusion

The features extracted using ResNet50, VGGish, and YAMNet produced high dimensionality (2048-D, 128-D, and 512-D, respectively) embeddings. Although these features carried rich information, they also introduced redundancy, noise, and computational burden, which could reduce the effectiveness of the classifier. To manage this issue, the proposed framework incorporated a feature selection stage to retain only the most discriminative components of the combined feature vectors. This step improved computational

efficiency and supported better generalization by removing irrelevant or weakly informative attributes. The study employed the SelectKBest method with the Analysis of Variance (ANOVA) F-test as the feature scoring criterion. The ANOVA F-test measured the statistical dependence between each feature and the class labels (real vs. fake speech). Features that showed stronger separation between the two classes received higher scores and were selected for the final representation. For a given feature f_j, the F-statistic is defined as:

$$F_j = \frac{\text{variance between classes of } f_j}{\text{variance within classes of } f_j}. \tag{16}$$

A higher F_j value indicated that a feature provided stronger separation between classes and therefore contributed more to classification. The study ranked features based on their F F-scores and selected the top k features. Experimental trials showed that $k = 3$ achieved a good balance between discriminative capability and computational efficiency. The determination of the number of features ($k = 3$) from each pre-trained model was based on an empirical analysis of the ANOVA F-test results. By ranking features according to their F-scores, it was observed that the top-3 features from ResNet50, VGGish, and YAMNet captured the most significant variance related to synthetic artifacts. While a higher k (e.g., $k = 10$ or $k = 50$) was tested, the incremental gain in classification accuracy was less than 0.3%, while the computational complexity for the final classifier increased. Therefore, the top-3 features were selected from each model to ensure a highly discriminative and optimized framework for high-speed, real-time detection. Feature selection produced two main benefits. First, it reduced computational demand because the classifier operated on a compact 3-dimensional input instead of very high-dimensional embeddings such as the 2048-dimensional output from ResNet50. Second, it acted as a regularization mechanism by guiding the classifier to focus on highly informative cues. The study applied feature selection separately to embeddings from ResNet50, VGGish, and YAMNet. This strategy ensured that each model contributed its most distinctive dimensions to the fusion stage and preserved complementary strengths. After selecting the top 3 features from each model and concatenating them, the fused feature space remained compact yet highly effective input for the classifier. Mathematically, the concatenated embeddings generated by the three fine-tuned networks is presented in Eq. (17):

$$F_{\text{hybrid}} = \left[\hat{f}_{\text{ResNet50}} \,||\, f_{\text{VGGish-tuned}} \,||\, f_{\text{YAMNet-tuned}}\right], \tag{17}$$

where || denotes concatenation.

3.5 Classification

The framework performed the final classification of speech samples into two categories: real or fake. This stage determined whether the processed embeddings represented authentic or synthetic voices. The study implemented a lightweight fully connected neural network for this classification. This architecture provided an effective balance between accuracy and efficiency compared to deeper classifiers, which require high computational resources. The network contained two hidden layers with dropout regularization, followed by a sigmoid output layer for binary classification. Formally, given an input feature vector $x \in \mathbb{R}^k$, where $k = 3$ after feature selection, the forward pass of the classifier can be expressed as:

$$h_1 = \phi(W_1 x + b_1), \tag{18}$$

$$h_1' = \text{Dropout}(h_1, p = 0.3), \tag{19}$$

$$h_2 = \phi(W_2 h_1' + b_2), \tag{20}$$

$$\hat{y} = \sigma(W_o h_2 + b_o), \tag{21}$$

where W_i and b_i are the learnable weights and biases of the i-th layer, $\phi(\cdot)$ denotes the Rectified Linear Unit (ReLU) activation, and $\sigma(\cdot)$ is the sigmoid function:

$$\sigma(z) = \frac{1}{1 + e^{-z}}. \tag{22}$$

The output $\hat{y} \in [0,1]$ represents the predicted probability of the input belonging to the fake class. A binary decision $\tilde{y}$ is made according to:

$$\tilde{y} = \begin{cases} 1 & \text{if } \hat{y} \geq 0.5 \quad (\text{fake}), \\ 0 & \text{if } \hat{y} < 0.5 \quad (\text{real}). \end{cases} \tag{23}$$

The classifier was trained using the binary cross-entropy (BCE) loss function:

$$\mathcal{L} = -\frac{1}{N} \sum_{i=1}^{N} \Big[y_i \log \hat{y}_i + (1 - y_i) \log(1 - \hat{y}_i) \Big], \tag{24}$$

where $y_i \in \{0,1\}$ is the ground truth label and $\hat{y}_i$ is the predicted probability for the i-th sample. To address the high dimensionality of the fused feature space, a systematic parameter search was conducted using 3-fold stratified cross-validation. Optimization was performed using the Adam optimizer with a learning rate of $\eta = 10^{-3}$ and a batch size of 256. The study evaluated the models using a stratified 70/15/15 train–validation-test split. To ensure robustness and prevent overfitting to specific dataset artifacts, a multi-stage dropout mechanism ($p_{drop1} = 0.4$, $p_{drop2} = 0.3$) was implemented within the dense layers. Furthermore, an early stopping criterion with a patience of 10 epochs was enforced. This ensured that the final model parameters represent the state of optimal generalizability on the validation manifold. The evaluation reported accuracy, confusion matrices, and precision, recall, and F1-score metrics. Visualization of confusion matrices and t-SNE provided additional insight into the strengths and weaknesses of each model. The pipeline of proposed methodology is also presented in Algorithm 1.

Algorithm 1: Hybrid deepfake voice detection

Require: Dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$, where i indexes the samples, N is dataset size, $y_i = 0$ for *real* and $y_i = 1$ for *fake*

Ensure: Predictions $\tilde{y}_i \in \{0,1\}$ for each sample

1: **Feature extraction:** $f_i^{(R)} \in \mathbb{R}^{2048}$ (ResNet50), $f_i^{(V)} \in \mathbb{R}^{128}$ (VGGish), $f_i^{(Y)} \in \mathbb{R}^{512}$ (YAMNet)

2: **Feature selection (ANOVA *F*):** for each set $f_i^{(m)}$ rank features by $F_j = \mathrm{Var}_{\mathrm{between}}(f_j)/\mathrm{Var}_{\mathrm{within}}(f_j)$; keep top-$k$ ($k = 3$) $\Rightarrow \tilde{f}_i^{(R)}, \tilde{f}_i^{(V)}, \tilde{f}_i^{(Y)}$

3: **Fusion:** $F_i = \left[\tilde{f}_i^{(R)} \parallel \tilde{f}_i^{(V)} \parallel \tilde{f}_i^{(Y)}\right] \in \mathbb{R}^9$

4: **Classifier (FCNN):** $\hat{y}_i = \sigma\big(W_o\, \phi\big(W_2\, \phi\big(W_1 F_i + b_1\big) + b_2\big) + b_o\big)$

5: **Decision:** $\tilde{y}_i \leftarrow \mathbb{1}[\hat{y}_i \geq 0.5]$

6: **Training objective:** $\mathcal{L} = -\frac{1}{N} \sum_{i=1}^{N} \big(y_i \log \hat{y}_i + (1 - y_i) \log(1 - \hat{y}_i)\big)$; optimize with Adam ($\eta = 10^{-3}$)

4 Result and Discussion

4.1 ResNet-50 Performance

This study used a fine-tuned ResNet-50 model as a deep feature extractor for deepfake audio detection. Instead of performing direct end-to-end classification, the network extracted high-level representations

and a feature selection strategy retained only three most discriminative features. These selected hybrid features were then provided to a conventional classification algorithm. This design reduced computational complexity and supported lightweight deployment. The confusion matrix in Fig. 3a demonstrates an overall classification accuracy of 64.7%, which indicates moderate discriminative performance. The model correctly identified a large proportion of real audio samples, while misclassification remains more frequent for fake audio. This imbalance suggests that the extracted feature subset captured the stable acoustic characteristics of genuine speech more effectively than subtle manipulation artifacts introduced in deepfake generation. Class-wise evaluation shown in Table 1 further clarifies this behavior. The fake class shows high precision (≈0.87) but low recall (≈0.34). The classifier therefore made reliable false predictions when it adheres to that label, yet it failed to detect many manipulated samples. In contrast, the real class achieves very high recall (≈0.95) but lower precision (≈0.59), which means that the model tends to favor the real class and incorrectly labels a considerable portion of fake samples as genuine. This bias reflects the difficulty of preserving fine-grained synthetic cues when reducing deep representations to only three hybrid descriptors. The t-SNE visualization of the learned feature space shown in Fig. 3b supports this interpretation. Although partial clustering appears between real and fake samples, the distributions overlap substantially. Fake samples scatter across the real region, which indicates that the selected features do not form a sharply separable manifold. The feature space therefore lacks a strong decision boundary, which directly contributes to the reduced recall for fake detection. The training and validation accuracy curves in Fig. 3c level-off at around 0.83–0.84. Loss curves decrease smoothly without major divergence. This pattern indicates effective transfer learning without overfitting. The disparity between good training performance and reduced testing accuracy, however, suggests that the major limitation is representational capacity following aggressive feature reduction, rather than feature optimization.

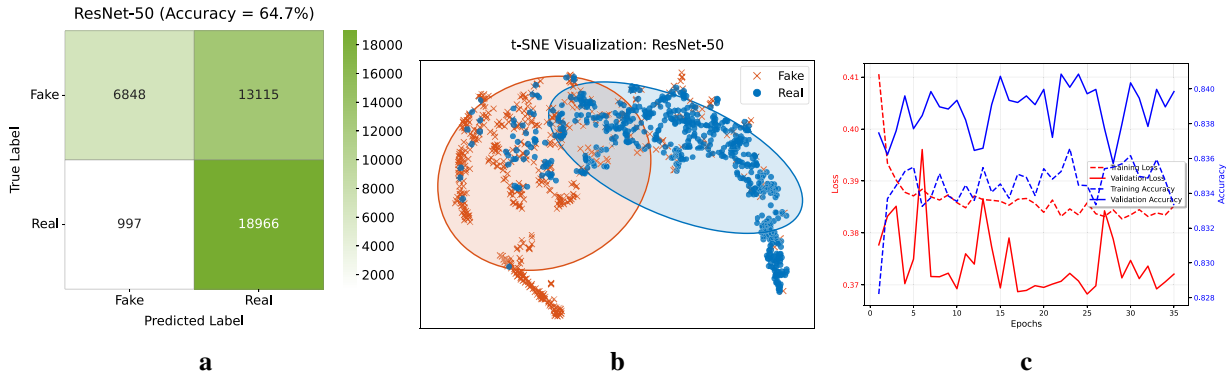

Figure 3: ResNet-50 results using selected features: (**a**) confusion matrix, (**b**) t-SNE feature visualization, and (**c**) training performance.

Table 1: Performance comparison of classification models.

Model	Precision	Recall	F1-Score	Accuracy
ResNet-50	0.73	0.65	0.61	0.65
VGGish	0.68	0.67	0.65	0.66
YAMNet	0.99	0.99	0.99	0.99
Hybrid	1.00	1.00	1.00	1.00

4.2 VGGNet Performance

In contrast to residual architectures, VGGNet adopts uniform convolutional structure which progressively aggregates the spectro-temporal information. VGGNet created high-level acoustic embeddings which were further processed to a small set of informative features and classified with a classifier. The confusion matrix in Fig. 4a shows that the overall accuracy is about 66.2%, which is slightly better than the ResNet-based feature extractor. The model exhibited good recognition of real speech with a high percentage of samples predicted correctly. However, there is a sizable number of instances of fake audio that are mislabeled as real. This shows that the extraction of features based on VGGNet being biased toward stable acoustic structure against an anomaly-sensitive that is generated by synthesis processes. Class-wise metrics in Table 1 further confirm this trend. The fake class achieved moderate levels of precision (≈0.74) but low recall (≈0.50). In comparison, the real class achieved high recall values (≈0.83) but lower precision (≈0.62) which points to a consistent tendency on real speech prediction. This reflects the domination of globally coherent acoustic patterns in the selected feature VGGNet subsets, while localized synthesis artifacts are not preserved sufficiently. The t-SNE projection in Fig. 4b suggests these interpretation is correct. Fake samples spread widely over regions which link to real audio. As a result, the classifier struggled to establish a discriminative boundary that reliably isolates fake audio. Training and validation curves in Fig. 4c demonstrate a stable convergence. Accuracy flattens out at approximately 0.66 for both training and validation set and loss curves stay very close together throughout training. This behavior confirms effective transfer learning and indicates that performance is not limited by overfitting or unstable optimization.

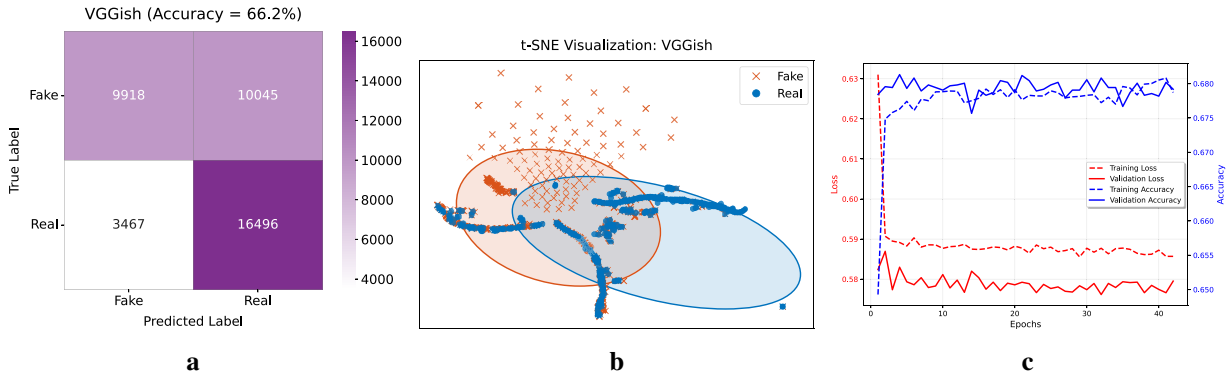

Figure 4: VGGNet results using selected features: (**a**) confusion matrix, (**b**) t-SNE feature visualization, and (**c**) training performance.

4.3 YAMNet Performance

Unlike general-purpose vision backbones adapted to spectrogram inputs, YAMNet is inherently designed for audio analysis and pre-trained on large-scale acoustic datasets. This architectural alignment enables the model to capture meaningful audio representations under aggressive feature reduction. The confusion matrix in Fig. 5a reveals near-perfect classification performance, with an overall accuracy of 99%. Both real and fake audio samples are classified with extremely high correctness, and only a negligible number of real audios are misclassified. However, a significant number of fake audio samples were misclassified as real, whereas the inverse occurred far less frequently. This result demonstrates YAMNet's strong capability to encode discriminative cues that reliably distinguish genuine speech from synthesized or manipulated audio. Class-wise performance metrics presented in Table 1 further confirm this robustness. Precision, recall, and F1-score for both classes approach unity, which indicates the balanced and unbiased predictions. This balanced behavior suggests that the extracted features preserve manipulation-specific artifacts and maintain

sensitivity to authentic speech characteristics. The t-SNE visualization shown in Fig. 5b provides a evidence of this discriminative power. Real and fake samples form two compact and well-separated clusters with minimal overlap. The tight intra-class grouping and large inter-class margin indicate that YAMNet learns a highly structured feature space in which deepfake artifacts are consistently encoded. Such separability directly facilitates the classifier's ability to establish a strong and stable decision boundary. Training dynamics illustrated in Fig. 5c demonstrate rapid and stable convergence. Training and validation accuracy approach saturation near 1.0 quickly, and loss curves reduce sharply. The lack of divergence between the training and validation curves indicates good generalization and no overfitting. This behavior highlights the effectiveness of transfer learning when the pretrained model is closely aligned with the target audio domain.

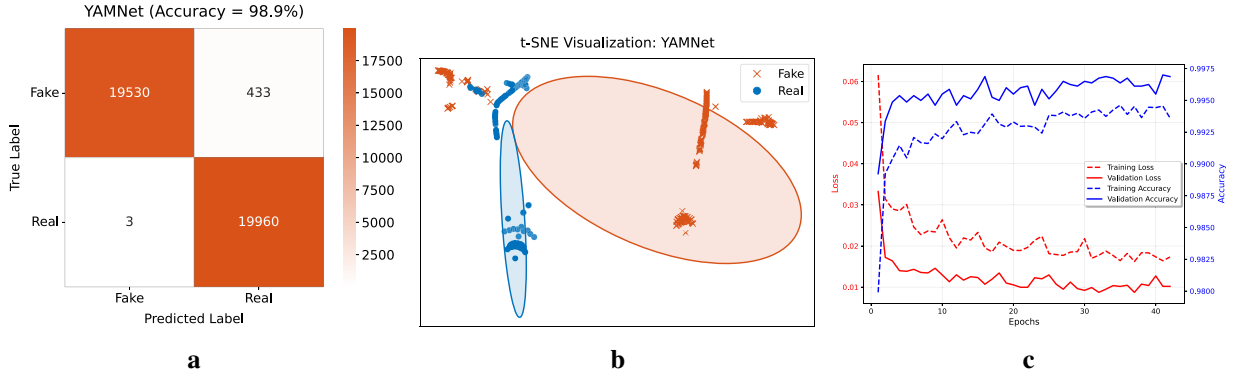

Figure 5: YAMNet results using selected features: (**a**) confusion matrix, (**b**) t-SNE feature visualization and (**c**) training performance.

4.4 Hybrid Model Performance

In this configuration, the framework integrated complementary deep representations extracted from three networks and kept a selected subset of features that was highly informative. This design leveraged the strengths of individual models and produced a robust and expressive representation for deepfake audio detection. Empirical results have shown a significant performance increase over all single backbones. The hybrid model achieved an overall accuracy of 99.7%, where classification errors reduced to a marginal level as shown in Fig. 6a. In particular, the missclassification rate fake audios is significantly lower compared to standalone models. This indicates that the fused feature space contains manipulation-specific cues that were previously suppressed by aggressive dimensionality reduction. Beyond overall accuracy, the hybrid model showed consistent behavior among classes. The achieved precision, recall and F1-score values remained uniform for both real and fake audio as given in Table 1 which confirms that the classifier is not biased in favor of one class over the other. The results show that the fused features effectively encoded both stable speech structure and subtle synthesis artifacts. This performance is further explained by the structure of the learned embedding space. As seen from the t-SNE projection in Fig. 6b, real and fake samples appear in two well-separated regions with a clear separation margin. Compared with previous models, overlap between classes is minimal, and intra-class dispersion is reduced. This means that feature fusion increased the discrimination and enabled the classifier to operate with a well-defined decision boundary. Accuracy quickly converged towards saturation both for training and validation data (Fig. 6c), while the corresponding loss values remained stable during training. The lack of divergence between training and validation curves implies good generalization and no overfitting. To evaluate the framework's robustness across varied decision thresholds, the Receiver Operating Characteristic (ROC) curves and Area Under the Curve (AUC) scores were computed for all experimental configurations (Fig. 7). The results demonstrate that the proposed hybrid

framework achieved a near-perfect AUC of 0.9999 while incorporating a more diverse, multi-scale feature set (Global, Semantic, and Local). Notably, the hybrid approach significantly outperforms the ResNet50 and VGGish individual models. The hybrid model curve toward the upper-left coordinate $(0,1)$ signifies an exceptional true positive rate (TPR) at extremely low false positive rates (FPR). This confirms the model's high reliability for digital forensic applications.

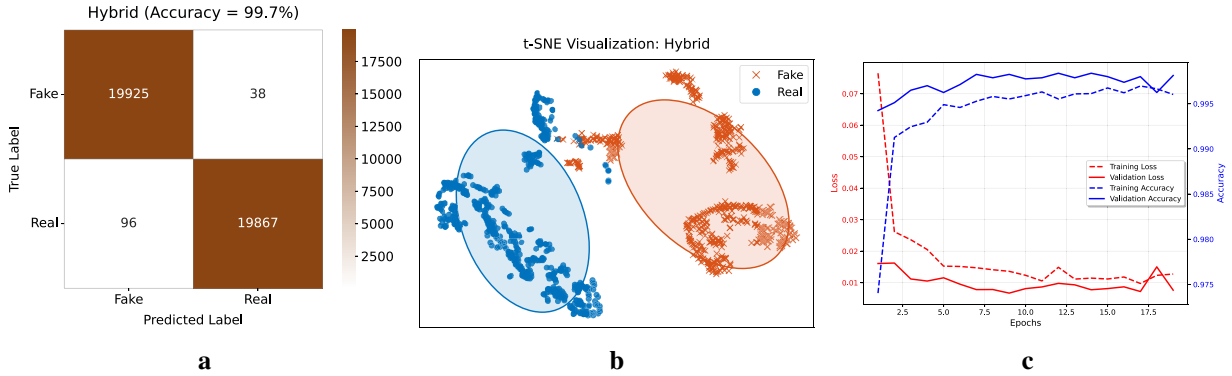

Figure 6: Hybrid results using selected features: (**a**) confusion matrix, (**b**) t-SNE feature visualization and (**c**) training performance.

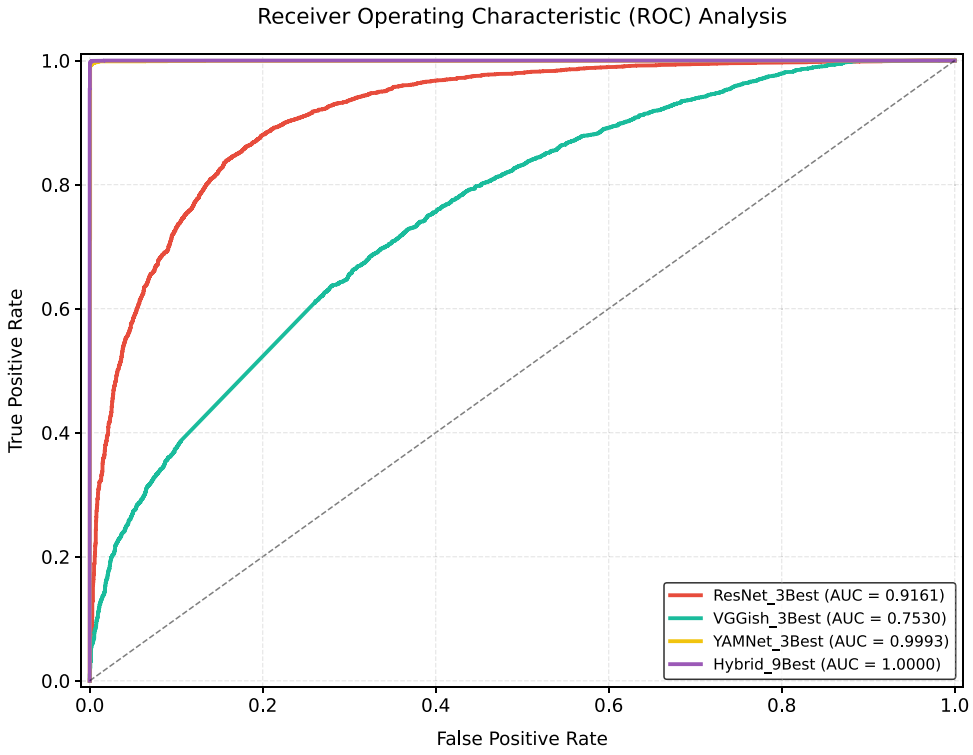

Figure 7: ROC analysis comparison of models with the hybrid framework.

4.5 Cross Dataset Validation

To evaluate the generalization capabilities of the proposed framework, a cross-dataset evaluation was performed using the Fake-or-Real (FoR) dataset [24]. The numerical results across all evaluated

architectures are summarized through confusion matrices and accuracy metrics. The proposed hybrid model demonstrated superior robustness, achieving a peak accuracy of 99.03%. As shown in the confusion matrix (Fig. 8), the hybrid model correctly identified 1383 fake samples and 1382 real samples, with only 27 total misclassifications. In comparison, the individual models exhibited lower generalization performance on the FoR dataset. The ResNet50 model achieved an accuracy of 98.14%, while the YAMNet and VGGish models showed significantly lower cross-dataset stability with accuracies of 82.74% and 71.99%, respectively. Specifically, while the hybrid model maintained a near-perfect balance, the VGGish model struggled with a high false-negative rate. These results indicate that the hierarchical fusion of heterogeneous embeddings effectively compensates for the domain-specific weaknesses of individual pre-trained models. The results on this dataset provide a highly reliable detection mechanism for various deepfake datasets.

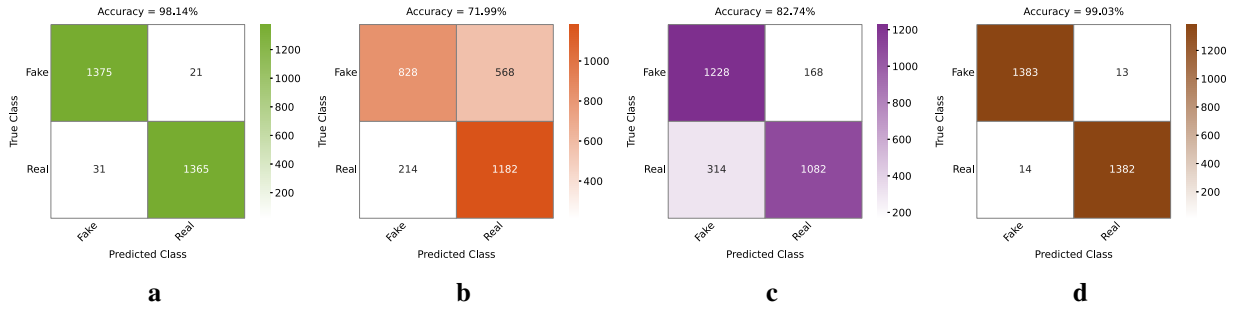

Figure 8: Proposed framework confusion metrics using FoR dataset: (**a**) ResNet-50, (**b**) VGGNet, (**c**) YAMNet and (**d**) hybrid model.

4.6 Comparative Analysis with Existing Studies

The performance of the proposed hybrid deepfake audio detection framework was rigorously evaluated against state-of-the-art baseline methods. As illustrated in Table 2, the proposed model achieved a superior overall 99.7% accuracy and outperformed the existing baseline approaches. A detailed examination of baseline models revealed several limitations that restrict their effectiveness in forensic-level deepfake detection. The BiLSTM method [25] achieved 98.8% accuracy using MFCCs and LFCCs descriptors. This model improved classification performance. However, its reliance on shallow statistical features limits the ability to encode complex spectro-temporal irregularities introduced during voice synthesis. The CNN-based approach [26] improved performance to 99.0% accuracy by using learned spectral representations. However, a single-stream convolutional model primarily captured global magnitude patterns and overlooked fine-grained phase distortions and micro-temporal inconsistencies characteristic of synthesized speech. This constraint led to class bias, where real audio samples were predicted more accurately than fake samples. Similarly, the MFCC-based framework [27] integrated handcrafted spectral features with deep learning models (LSTM, VGG16) and achieved 93.0% accuracy. Although the framework improved the features diversity, the approach remained dependent on handcrafted descriptors and did not explicitly used domain-aligned audio representations learned through large-scale pretraining. In contrast, the proposed framework integrated diverse pretrained models to capture manipulation-sensitive acoustic patterns rather than relying on spectral textures. The fusion of spatial, semantic, and acoustic-event representations reduced architectural bias and enhanced feature diversity. Furthermore, the discriminative feature selection phase retained the most informative dimensions and maintained lightweight deployment feasibility. The effectiveness of this technique is presented in the experimental results. The proposed method achieved balanced recall for both fake and real audio classes and a near-perfect F1-score. These results shows that the proposed method eliminated the common trade-off between fake recall and real recall observed in many deepfake

detection studies. Therefore, the proposed framework showed superiority in terms of model complexity, representation diversity, domain-specific pretraining, and targeted feature fusion. These factors ensured improved robustness and reliability for real-world deepfake audio forensic applications compared to existing baseline methods.

Table 2: Performance comparison between existing studies and proposed hybrid framework.

Study	Dataset	Technique	Feature Type	Model Type	Accuracy (%)
[25]	In-the-Wild	Bidirectional Long Short-Term Memory	MFCCs + LFCCs	Deep Learning	98.8
[26]	Fake-or-Real	CNN Model	Spectrogram-based Deep Features	Single CNN	99.0
[28]	Fake-or-Real	Pre-trained feature embeddings	Original + injected cues	Ensemble CNN	94.6
[27]	Fake-or-Real	MFCC + Spectral Descriptors	Engineered Features	LSTM/VGG16	93.0
This	In-the-Wild Fake-or-Real	Multi-stream Transfer Learning	Deep Audio + Spatial + Semantic Fusion	Hybrid + Feature Selection	99.7 99.3

To address the potential risk of information loss due to dimensionality reduction, a sensitivity analysis was conducted to evaluate the impact of the parameter k (the number of features selected per model) on detection performance. As illustrated in Table 3, an increase in the total hybrid feature count from 9 (k = 3) to 30 (k = 10) resulted in a marginal accuracy improvement of only 0.07% (from 99.87% to 99.94%). Further increase of feature count to 150 resulted in a slight performance degradation to 99.57%, likely due to the inclusion of redundant or noisy features. These results demonstrate that the top-3 features from each architecture captured the most critical non-linear artifacts. Moreover, results also validates that the k = 3 achieved the optimal threshold for state-of-the-art accuracy with maximum computational efficiency.

Table 3: Accuracy across different feature counts (k).

Feature Set	k per Model	Total Features	Accuracy (%)
9-best Hybrid	3	9	99.87%
30-best Hybrid	10	30	99.94%
150-best Hybrid	50	150	99.57%

5 Conclusion

This study presented a robust hybrid framework to detect deepfake voices. Three different models, ResNet50, VGGish, and YAMNet, are combined to effectively capture diverse acoustic features. The experimental results demonstrated that using models like ResNet50 and VGGish provides a strong foundation for spectral analysis. However, their performance is significantly enhanced when fused with audio-native models such as YAMNet. The implementation of a SelectKBest feature selection strategy proved vital,

which allowed the model to achieve near-perfect detection accuracy while maintaining a lightweight feature space suitable for real-world, resource-constrained deployment. The hybrid model achieved 99.7% accuracy and outperformed many baseline methods. Most models struggle with bias and tend to favor real speech. However, this study achieved a perfect 1.00 F1-score for both real and fake audios. This proves that YAMNet is especially good at spotting the tiny timing and phase errors found in synthetic voices. While the current framework showed exceptional performance, future research will incorporate deep speaker embedding models such as X-vector, ECAPA-TDNN, and ResNet-TDNN to capture identity-based anomalies. The integration of these specialized models could provide a more holistic biometric analysis. Additionally, self-supervised learning will be explored to further enhance the model's ability to generalize to emerging speech synthesis technologies.

Acknowledgement: Not applicable.

Funding Statement: This work is supported by Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia through the Researchers Supporting Project PNURSP2026R333.

Author Contributions: The authors confirm contribution to the paper as follows: formal analysis, investigation, methodology, and writing—original draft preparation, Rashid Jahangir; data curation, resources, software, visualization, writing—review and editing, funding acquisition: Nazik Alturki; methodology, conceptualization, visualization: Muhammad Zubair Khan. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: Data openly available in a public repository.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Li M, Ahmadiadli Y, Zhang XP. A survey on speech deepfake detection. ACM Comput Surv. 2025;57(7):1–38. doi:10.1145/3714458.
2. Pham L, Lam P, Tran D, Tang H, Nguyen T, Schindler A, et al. A comprehensive survey with critical analysis for deepfake speech detection. Comput Sci Rev. 2025;57(4):100757. doi:10.1016/j.cosrev.2025.100757.
3. Wang X, Delgado H, Tak H, Jung JW, Shim HJ, Todisco M, et al. ASVspoof 5: crowdsourced speech data, deepfakes, and adversarial attacks at scale. In: The Automatic Speaker Verification Spoofing Countermeasures Workshop (ASVspoof 2024). Kos Island, Greece: ISCA; 2024. p. 1–8. [cited 2026 Feb 12]. Available from: https://hal.science/hal-04803294.
4. Zhang B, Cui H, Nguyen V, Whitty M. Audio deepfake detection: what has been achieved and what lies ahead. Sensors. 2025;25(7):1989. doi:10.3390/s25071989.
5. Hershey S, Chaudhuri S, Ellis DP, Gemmeke JF, Jansen A, Moore RC, et al. CNN architectures for large-scale audio classification. In: Proceedings of the 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP); 2017 Mar 5–9; New Orleans, LA, USA. New York, NY, USA: IEEE; 2017. p. 131–5.
6. Dua M, Joshi S, Dua S. Data augmentation based novel approach to automatic speaker verification system. e-Prime-Adv Electr Eng Electron Energy. 2023;6(5):100346. doi:10.1016/j.prime.2023.100346.
7. Mahum R, Irtaza A, Javed A, Mahmoud HA, Hassan H. DeepDet: YAMNet with BottleNeck attention module (BAM) for TTS synthesis detection. EURASIP J Audio Speech Music Process. 2024;2024(1):18.
8. Abdulhamied RM, Naiem S, Nasr MM, Moussa FA. Deepfake audio detection using feature-based and deep learning approaches: ANN vs. ResNet50. Int J Adv Comput Sci Appl. 2025;16(6):316–23.
9. Souza LMD, Guido RC, Contreras RC, Viana MS, Bongarti MADS. Improving voice spoofing detection through extensive analysis of multicepstral feature reduction. Sensors. 2025;25(15):4821. doi:10.3390/s25154821.

10. Van Den Oord A, Dieleman S, Zen H, Simonyan K, Vinyals O, Graves A, et al. Wavenet: a generative model for raw audio. arXiv:1609.03499. 2016.
11. Shen J, Pang R, Weiss RJ, Schuster M, Jaitly N, Yang Z, et al. Natural tts synthesis by conditioning wavenet on mel spectrogram predictions. In: Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP); 2018 Apr 15–20; Calgary, AB, Canada. New York, NY, USA: IEEE; 2018. p. 4779–83.
12. Oord A, Li Y, Babuschkin I, Simonyan K, Vinyals O, Kavukcuoglu K, et al. Parallel wavenet: fast high-fidelity speech synthesis. In: Proceedings of the International Conference on Machine Learning; 2018 Jul 15–20; Stockholm, Sweden. Cambridge, CA, USA: PMLR; 2018. p. 3918–26.
13. Arık SÖ, Chrzanowski M, Coates A, Diamos G, Gibiansky A, Kang Y, et al. Deep voice: real-time neural text-to-speech. In: Proceedings of the International Conference on Machine Learning; 2017 Aug 6–11; Sydney, NSW, Australia. Cambridge, CA, USA: PMLR; 2017. p. 195–204.
14. Toda T, Black AW, Tokuda K. Voice conversion based on maximum-likelihood estimation of spectral parameter trajectory. IEEE Trans Audio Speech Lang Process. 2007;15(8):2222–35.
15. Kameoka H, Kaneko T, Tanaka K, Hojo N. Stargan-VC: non-parallel many-to-many voice conversion using star generative adversarial networks. In: Proceedings of the 2018 IEEE Spoken Language Technology Workshop (SLT); 2018 Dec 18–21; Athens, Greece. New York, NY, USA: IEEE; 2018. p. 266–73.
16. Yi J, Bai Y, Tao J, Ma H, Tian Z, Wang C, et al. Half-truth: a partially fake audio detection dataset. In: Proceedings of the Interspeech 2021; 2021 Aug 30–Sep 3; Brno, Czech Republic. p. 1654–8.
17. Wang X, Yamagishi J, Todisco M, Delgado H, Nautsch A, Evans N, et al. ASVspoof 2019: a large-scale public database of synthesized, converted and replayed speech. Comput Speech Lang. 2020;64(2):101114. doi:10.1109/tbiom.2021.3059479.
18. Baevski A, Zhou Y, Mohamed A, Auli M. wav2vec 2.0: a framework for self-supervised learning of speech representations. Adv Neural Inf Process Syst. 2020;33:12449–60.
19. Hsu WN, Bolte B, Tsai YHH, Lakhotia K, Salakhutdinov R, Mohamed A. Hubert: self-supervised speech representation learning by masked prediction of hidden units. IEEE/ACM Trans Audio Speech Lang Process. 2021;29:3451–60. doi:10.1109/taslp.2021.3122291.
20. Xie Z, Li B, Xu X, Liang Z, Yu K, Wu M. FakeSound: deepfake general audio detection. In: Proceedings of the Interspeech 2024; 2024 Sep 1–5; Kos Island, Greece. p. 112–6.
21. Radford A, Kim JW, Xu T, Brockman G, McLeavey C, Sutskever I. Robust speech recognition via large-scale weak supervision. In: Proceedings of the International Conference on Machine Learning; 2023 Jul 23–29; Honolulu, HI, USA. Cambridge, CA, USA: PMLR; 2023. p. 28492–518.
22. Zhu Y, Koppisetti S, Tran T, Bharaj G. Slim: style-linguistics mismatch model for generalized audio deepfake detection. Adv Neural Inf Process Syst. 2024;37:67901–28. doi:10.52202/079017-2168.
23. Müller N, Czempin P, Diekmann F, Froghyar A, Böttinger K. Does audio deepfake detection generalize? In: Proceedings of the Interspeech 2022; 2022 Sep 18–22; Incheon, Republic of Korea. p. 2783–7.
24. Reimao R, Tzerpos V. FoR: a dataset for synthetic speech detection. In: Proceedings of the 2019 International Conference on Speech Technology and Human-Computer Dialogue (SpeD); 2019 Oct 10–12; Timisoara, Romania. p. 1–10.
25. Momu SA, Siddiqui RR, Shanto SS, Ahmed Z. A comprehensive approach to deepfake audio detection: using feature fusion and deep learning. In: Proceedings of the 2024 27th International Conference on Computer and Information Technology (ICCIT); 2024 Dec 20–22; Cox's Bazar, Bangladesh. New York, NY, USA: IEEE; 2024. p. 351–6.
26. Valente LP, de Souza MM, Da Rocha AM. Speech audio deepfake detection via convolutional neural networks. In: Proceedings of the 2024 IEEE International Conference on Evolving and Adaptive Intelligent Systems (EAIS); 2024 May 23–24; Madrid, Spain. New York, NY, USA: IEEE; 2024. p. 1–6.
27. Hamza A, Javed ARR, Iqbal F, Kryvinska N, Almadhor AS, Jalil Z, et al. Deepfake audio detection via MFCC features using machine learning. IEEE Access. 2022;10(8):134018–28. doi:10.1109/access.2022.3231480.
28. Li Y, Lyu C, Wang L, Luo W, Zhang K, Schuller BW. The affective bridge: unifying feature representations for speech deepfake detection. arXiv:2512.11241. 2025.