University of Dayton

Securing Voice Assistants Against Deepfake Threats:

Real-Time Detection & Privacy Protection

Name: Sai Woon Tip

Department: Computer Science

Student ID: 101821675

Advisor: Tasnia Ashrafi Heya

Table of Content

Abstract

Voice assistants (VAs) have become an integral part of modern digital life, yet their increasing sophistication is paralleled by their vulnerability to advanced deepfake audio attacks. These attacks, where synthetically generated voices issue malicious commands, pose significant security and privacy risks. The rapid advancement of generative AI necessitates the development of robust countermeasures. This research project presents a comparative study to evaluate the effectiveness of different machine learning methodologies for detecting deepfake voice commands. The approach involved two key stages. First, a specialized dataset of 192 audio clips, comprising both authentic human speech and synthetic commands generated by four distinct Text-to-Speech engines. Second, we trained and systematically evaluated two categories of models: classical machine learning algorithms (e.g., Random Forest) using extracted acoustic features, and deep learning models (e.g., AlexNet, ResNet) using mel-spectrograms as image-based input. The primary contribution of this work is the direct performance comparison of these two distinct approaches, providing valuable insights for model selection in this domain. Our results indicate that while both methods achieve high accuracy, classical models using a feature-based approach demonstrated exceptional performance, with models reaching 97.9% accuracy on the test set. This study establishes a foundational methodology and provides a crucial benchmark for future work in developing robust security solutions for voice-based interfaces.

1. Introduction

1.1. Background

In the last decade, voice assistants (VAs) such as Amazon's Alexa, Google Assistant, and Apple's Siri have transitioned from technological novelties to deeply integrated components of daily life. The global augmentation of these systems is staggering, with estimates suggesting the number of voice assistants in use reached 8.4 billion by 2024, a figure that outnumbers the world's population [21]. This market is projected to further grow into a multi-billion-dollar industry, driven by the integration of VAs into smart home devices, consumer electronics, and enterprise solutions. Users now rely on voice commands for a vast range of tasks, from information retrieval and smart home control to accessing sensitive financial and healthcare information.

Parallel to the growth of VAs, generative artificial intelligence has seen exponential advancement. Deepfake technology, a product of this progress, leverages deep learning to create highly realistic synthetic media [1]. While often associated with video, audio deepfakes present a particularly insidious threat. Modern Text-to-Speech (TTS) and Voice Conversion (VC) systems can generate artificial speech that is nearly indistinguishable from that of a real person, often requiring only a few seconds of a target's voice to create a convincing clone [20]. The increasing sophistication and accessibility of these tools, from open-source models to commercial services, have democratized the ability to create high-fidelity audio clones [29].

1.2. Motivation and Problem Statement

The convergence of voice assistants and accessible deepfake technology has created a critical security vulnerability. The threat model is no longer theoretical, with tangible risks to security, privacy, and the foundation of digital trust.

Security Threats: Malicious actors can exploit deepfake audio to issue unauthorized commands to VAs, such as "unlock the front door" or "transfer money". Such attacks can bypass even robust security measures like Automatic Speaker Verification (ASV). High-quality voice clones can successfully mimic a target user's unique voiceprint, leading to a more than five-fold increase in false acceptance rates [9, 24]. Furthermore, studies have shown that "partial fake" attacks, where a small cloned segment is inserted into genuine audio, can deceive commercial speaker recognition systems with a success rate as high as 95% to 97% [15].

Privacy Invasion: Attackers can use a mimicked voice to query a VA for sensitive personal information, such as "read my last messages" or "what is on my calendar today," thereby gaining unauthorized access to a user's private data.

This leads to the formal problem statement: Given the threat of deepfake audio attacks on voice assistants, there is a need for a systematic evaluation of detection models to identify effective and reliable methodologies. The lack of comparative benchmarks on challenging, modern datasets leaves a critical gap in understanding which approaches are best suited to distinguish between legitimate and synthetic voice commands.

1.3. Research Questions

This research is guided by a primary question and a set of supporting sub-questions designed to investigate the problem systematically.

Primary Question: Which machine learning methodology - a classical approach using engineered acoustic features or a deep learning approach using visual audio representations is more effective for detecting deepfake voice commands?

Sub-Questions:

- How does a purpose-built dataset, featuring synthetic audio from multiple state-of-the-art TTS engines, serve as an effective tool for training and evaluating detection models?
- What is the performance (in terms of accuracy, precision, recall, and F1-score) of various classical machine learning models when trained on a curated set of acoustic features?
- How does the performance of these classical models compare to that of common Convolutional Neural Network (CNN) architectures trained on mel-spectrograms derived from the same audio?

1.4. Scope and Objectives

The scope of this project is to conduct a comparative evaluation of machine learning techniques for deepfake voice command detection. The key objectives are:

- **Dataset Curation:** To generate and curate a balanced audio dataset consisting of real and deepfake voice commands, incorporating multiple human speakers and four distinct deepfake generation models.
- **Model Training and Comparison:** To train and evaluate two separate classes of models: classical machine learning classifiers using a set of engineered acoustic features, and deep learning CNN classifiers using mel-spectrogram images.

- **Performance Evaluation:** To rigorously evaluate and compare the performance of all models using key classification metrics, including **accuracy, precision, recall, and F1-Score.**
- **Comparative Analysis:** To analyze the results to determine the relative strengths and weaknesses of each modeling approach for this specific detection task, providing a benchmark for future research.

2. Literature Review

2.1. The Evolution of Deepfake Audio Generation

The ability to generate realistic synthetic speech has evolved rapidly, driven by advancements in deep learning architectures.

Autoregressive Models: Early breakthroughs in high-fidelity audio generation came from autoregressive models like DeepMind's WaveNet [18] and SampleRNN. These models generate audio waveforms one sample at a time, conditioning each new sample on the preceding ones. While capable of producing very high-quality audio, their sequential nature makes generation computationally intensive.

Generative Adversarial Networks (GANs): To accelerate generation, researchers turned to GANs. Models like WaveGAN adapt the GAN framework to the audio domain, where a generator network learns to produce realistic waveforms by competing against a discriminator network that tries to distinguish fakes from real audio.

Flow-based and Diffusion Models: More recent techniques have further improved both the speed and stability of audio generation. Flow-based models like WaveGlow and diffusion models like

WaveGrad offer parallel, non-autoregressive generation, significantly reducing synthesis time while maintaining high fidelity.

Voice Conversion (VC): A critical technology for impersonation attacks is Voice Conversion. VC models aim to transform the voice of a source speaker to sound like a target speaker while preserving the linguistic content. Techniques like StarGAN-VC enable many-to-many voice conversion without requiring parallel data, making it a powerful and flexible tool for creating deepfake identities [8].

2.2. State-of-the-Art in Deepfake Audio Detection

As generation techniques have advanced, so too have methods for detection.

Feature Engineering Approaches: Early detection methods relied on hand-crafted acoustic features to capture artifacts of the synthesis process. These include Mel-Frequency Cepstral Coefficients (MFCCs), Linear Prediction Cepstral Coefficients (LPCCs), and Constant Q Cepstral Coefficients (CQCCs), which can reveal unnatural spectral patterns not present in genuine human speech [26].

Spectrogram Analysis: A common approach is to convert audio signals into spectrograms, visual time-frequency representations, and treat the detection problem as an image classification task. Subtle inconsistencies in harmonics or phase, often imperceptible to the human ear, can become apparent in a spectrogram, providing discriminative features for a machine learning model [26].

Deep Learning Models for Detection: Modern detectors predominantly use deep learning. Convolutional Neural Networks (CNNs) are highly effective at analyzing spectrograms to find spatial patterns and artifacts. Recurrent Neural Networks (RNNs) and their variants, like Long Short-Term Memory (LSTM) networks, excel at capturing temporal dependencies. Hybrid models,

such as a Convolutional Recurrent Neural Network (CRNN), combine these strengths. Leading models like AASIST and RawNet2 have demonstrated strong performance on academic benchmarks [29].

Public Datasets and the Generalization Crisis: The primary benchmark for this field has been the ASVspoof (Automatic Speaker Verification Spoofing and Countermeasures) challenge series [10, 14]. While invaluable, these datasets have significant limitations for real-world VA security. They are often restricted to a single language (typically English), consist of clean, high-quality audio, and focus on speaker verification rather than command-and-control interactions [29]. This has led to a "generalization crisis," where models that excel on ASVspoof fail dramatically when exposed to the noise, compression, and linguistic diversity of real-world audio [6, 23]. For example, recent benchmarks like VoiceWukong have shown that top detectors' Equal Error Rates (EER) can skyrocket from under 1% on academic data to over 13-20% on in-the-wild samples [29].

2.3. Identifying the Research Gap

The literature demonstrate that while highly powerful deepfake detection models like AASIST and RawNet2 exist, their development and evaluation are focused on achieving state-of-the-art performance on large-scale, pre-established benchmarks like ASVspoof. This focus on pushing the boundaries of deep learning performance, however, leaves a specific methodological gap that this research aims to address. A key aspect of this gap is the context of the data itself. The performance of models on existing benchmarks like ASVspoof, which often contain longer-form declarative sentences, does not guarantee their effectiveness in the context of voice assistants. There is a need for benchmarks focused specifically on the short, imperative phrases ("Open YouTube," "Play music") that characterize VA interactions.

Therefore, this project fills a precise methodological niche. Instead of attempting to build a new state-of-the-art model, this research provides a foundational, comparative benchmark**.** It seeks to answer a practical question: In the specific context of securing VA commands against modern, accessible deepfake threats, is it possible to determine the deepfakes using machine learning module to filter against different spoofing attacks?

3. Experimental Setup

3.1 Data Collection Method

A custom dataset was created to train and evaluate the deepfake voice detection. The dataset consists of **192 audio clips**, equally divided into two classes: "**Bonifide**" (96 samples) and "**Deepfake**" (96 samples). The foundation for all recordings is a corpus of 16 common voice assistant commands, including phrases like "Open YouTube?" and "Play some music."

The authentic voice data was collected from **6 human participants** (4 male and 2 female) in a quiet office environment. Each participant recorded all 16 commands, generating a total of 96 authentic audio files. The recordings were captured using a **ReSpeaker 4-Mic Array USB device** and managed by a custom Python script.

To create a challenging and diverse set of synthetic voices, the 96 deepfake samples were generated using four different state-of-the-art Text-to-Speech (TTS) engines. This multi-generator approach prevents the model from overfitting to the audio artifacts of a single system. The same 16 command phrases were used to generate the synthetic data, with the following distribution:

- **OpenAI TTS:** 32 samples, using two distinct female voices (16 phrases each).
- **Google Gemini TTS:** 32 samples, using two distinct male voices (16 phrases each).
- **Microsoft Edge Read Aloud:** 16 samples, using its default natural-sounding voice.
- **Google Translate TTS (gTTS):** 16 samples.

The resulting dataset is balanced, to ensure an equal number of authentic and deepfake samples for each of the 16 commands.

3.2 Data Processing

After the initial data collection, the entire set of 192 audio files underwent a critical preprocessing phase to ensure uniformity and consistency. This was accomplished using a dedicated Python script (1_standardization.py) that leveraged the **pydub** library, a powerful tool for audio manipulation. The script applied the same transformations to every authentic and deepfake audio clip.

The standardization process involved several key steps within the script. First, each audio file was loaded into an AudioSegment object. Then, three methods were applied:

1. **Channel Conversion:** The set_channels(1) method was used to convert the audio to a **single-channel (mono)** format.
2. **Resampling:** The set_frame_rate (16000) method was used to standardize the sample rate of every file to a consistent **16,000 Hz**.
3. **Amplitude Normalization:** The loudness of each file was normalized to a target level of **-20 dBFS** (decibels relative to full scale).

This complete procedure is essential for eliminating technical variations in format and loudness, ensuring that the model learns from the intrinsic acoustic properties of the speech itself.

The processed files were then organized into two directories, real and fake, corresponding to their class. Each file was named using a systematic convention (class_speaker/sourceID_commandID.wav) for clear and systematic identification.

3.2.1 Features Extraction

Following preprocessing, a set of acoustic features was extracted from each audio file to create a structured dataset for machine learning. This process was executed using a Python script that leverages the **Librosa** library for audio analysis. To maintain consistency and manage processing time, a maximum duration of **5 seconds** from the beginning of each audio file was used for the analysis.

For each audio clip, a suite of features designed to capture different acoustic properties was computed. These included:

- **Mel-Frequency Cepstral Coefficients (MFCCs):** The first 20 coefficients, which are standard for representing the timbral characteristics of speech.
- **Chroma Frequencies:** Features that capture the harmonic and melodic content of the audio.
- **Spectral Features:** Including **Spectral Centroid** (indicative of the sound's "brightness"), **Spectral Bandwidth** (the range of frequencies), and **Spectral Rolloff** (the frequency below which a specified percentage of the total spectral energy lies).
- **Zero-Crossing Rate:** The rate at which the signal changes from positive to negative, often correlated with the presence of noise or percussive sounds.
- **Harmonic-Percussive Separation:** The audio was decomposed into its harmonic (tonal) and percussive (transient) components, and features were extracted from each.

To create a fixed-size feature vector for each audio file, the **mean** and **standard deviation** of each of these feature sets were calculated over the 10-second duration. The final output of this stage

was a single **features.csv** file. In this file, each row represents a single audio clip, and the columns contain its unique filename, its label ('real' or 'fake'), and the corresponding aggregated feature values.

3.2.2 Data Visualization

In addition to extracting numerical features, the audio data was transformed into a visual format for two primary purposes: 1) for exploratory analysis to identify discernible patterns between authentic and deepfake audio, and 2) to serve as image-based input for Convolutional Neural Network (CNN) models. This transformation was accomplished using dedicated Python scripts and the **Librosa** library.

Two types of spectrograms were generated from each preprocessed audio file:

1. **Linear Spectrograms:** These provide a direct visual representation of the spectrum of frequencies in the audio signal over time. They are primarily useful for qualitative analysis, allowing for the inspection of potential artifacts or structural differences that might distinguish real speech from synthetic speech.

2. **Mel-Spectrograms:** These are a perceptually-relevant variation of standard spectrograms where the frequency axis is mapped to the **mel scale**. The mel scale better reflects human pitch perception, making these representations a powerful input for machine learning models. The generated mel-spectrogram images were the primary data format used to train and evaluate the deep learning classifiers in the subsequent modeling stage.

Real audio

Fake audio

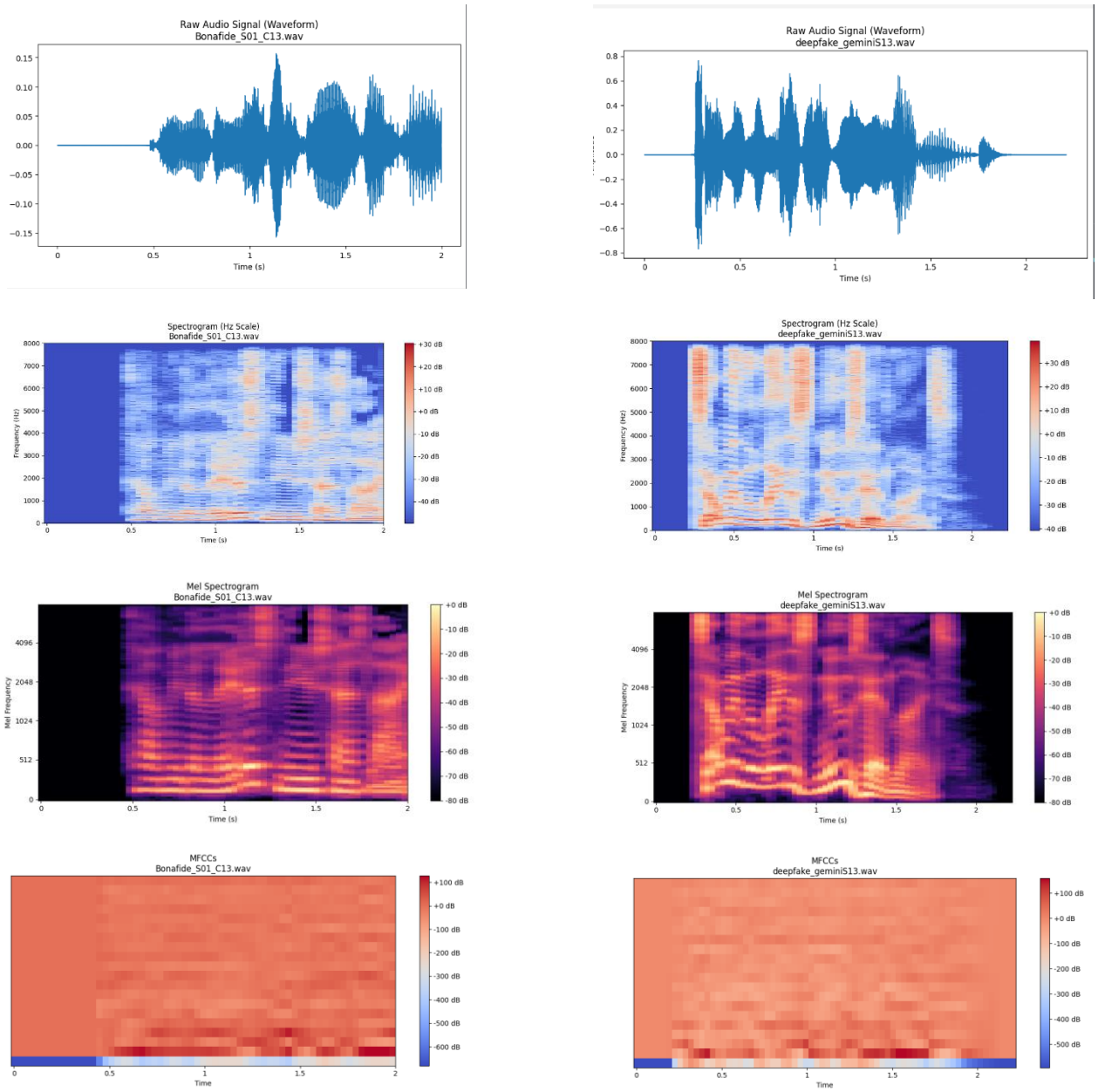

Figure 1: Visual Comparison of a 'Bonafide' and 'Deepfake(gemini)' Samples

3.3 Model Training

To identify the most effective method for detecting deepfake voice commands, a comparative study was conducted using two distinct modeling strategies: one employing classical machine learning algorithms on aggregated acoustic features, and another using deep learning models on visual representations of the audio.

3.3.1 Classical Machine Learning Approach

This approach evaluated a wide range of classical machine learning models on the structured dataset derived from features.csv. The workflow was carefully designed to prevent data leakage during preprocessing.

First, the dataset was partitioned into a **75% training set** and a **25% testing set consisting of 48 audio samples**, using stratification to maintain the class distribution in both sets. Following this split, feature scaling was performed. A **Min-Max Scaler** was fitted **exclusively on the training data**, and this fitted scaler was then used to transform both the training and testing sets to a uniform range of [0, 1]. This ensures that no information from the test set influenced the scaling process. Categorical labels were also converted to integers using a label encoder.

A suite of five models was then systematically trained on the processed training data and evaluated on the unseen test data:

- **Tree-Based Model:** Random Forest
- **Kernel-Based Models:** Support Vector Machine (SVM)
- **Neural Networks:** Multi-layer Perceptron (MLP)
- **Ensemble (Boosting) Models:** XGBoost, XGBoost Random Forest

Deep Learning Approach: This approach treated deepfake detection as an image classification problem, using the mel-spectrograms as direct input. This allows the models to learn discriminative features automatically.

- **Models Used:** Three different Convolutional Neural Network (CNN) architectures were trained and compared:
    - **AlexNet**
    - **4 different ResNet models**
    - **MobileNet v2**
- **Training Parameters:** The deep learning strategy employed a **transfer learning** methodology to leverage the power of pre-trained Convolutional Neural Networks (CNNs).
- The mel-spectrogram image dataset was partitioned into a **70% training set** and a **30% testing set**. To improve model generalization, **data augmentation** techniques (random horizontal flips and rotations) were applied exclusively to the training images.
    - **Image Size:** All mel-spectrogram images were resized to **224x224 pixels**. This is a standard input dimension for many well-known CNN architectures, particularly those pre-trained on the ImageNet dataset. Using a fixed size is a requirement for the network's input layer and ensures that all data points are processed uniformly.
    - **Batch Size:** A batch size of **16 and 32** were used. This parameter defines the number of training samples the model processes before its internal weights are

updated. The chosen size represents a trade-off between computational memory requirements and the stability of the learning process.

- **Epochs:** The models were trained for 45 epochs. An epoch is one complete pass through the entire training dataset. The number of epochs was determined by observing the model's performance on the validation set, aiming to train long enough for the model to learn the underlying patterns (**convergence**) without memorizing the training data (**overfitting**).
- **Loss Function & Optimizer: Binary Cross-Entropy** was used as the loss function, as it is specifically designed for binary (two-class) classification problems. The **Adam** optimizer was chosen to update the network's weights due to its adaptive learning rate capabilities and proven effectiveness in training deep networks.

4. Results

4.1 Evaluation Metrics

The performance of each model was assessed using standard classification metrics derived from the model's predictions on the test set. For this project, the "positive" class is 'fake' (as the goal is to detect fakes) and the "negative" class is 'real'.

The core components of the evaluation are:

- **True Positive (TP):** The model correctly identifies a **deepfake** audio clip as **'fake'**. This is a successful detection.
- **True Negative (TN):** The model correctly identifies a **real** audio clip as **'real'**. This is a successful rejection of a legitimate command.
- **False Positive (FP):** The model incorrectly identifies a **real** audio clip as **'fake'**. This is a **Type I error**, where a legitimate user is blocked.
- **False Negative (FN):** The model incorrectly identifies a **deepfake** audio clip as **'real'**. This is a **Type II error**, where an attack is missed.

Accuracy: The overall percentage of predictions that were correct. It is a good general measure but can be misleading on imbalanced datasets.

$$Accuracy = \frac{True\ Positives + True\ Negatives}{Total\ Predictions}$$

Precision: Of all the predictions made for a class, how many were correct. High precision for the 'fake' class means a low false positive rate.

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives}$$

Recall (Sensitivity): Of all the actual instances of a class, how many were correctly identified. High recall for the 'fake' class means the model is good at catching deepfakes.

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives}$$

F1-Score: The harmonic means of Precision and Recall. It provides a single score that balances both concerns, making it an excellent metric for evaluating overall performance.

$$F1\ score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

4.2 Testing Results of Classic Machine Learning Models

The five classical machine learning models were trained on 75% of the feature dataset and evaluated on the remaining 25% test set, which consisted of 48 audio samples.

Models	Average Accuracy	Precision		F1-score	
		Real	Fake	Real	Fake
RandomForest Classifier	97.9%	96%	100%	98%	98%
Support Vector Machine (SVM)	87.5%	85%	91%	88%	87%
MLP Classifier (Neural Nets)	97.9%	96%	100%	98%	98%

XGBClassifier	93.7%	89%	100%	94%	93%
XGBRFClassifier	93.7%	89%	100%	94%	93%

Fig 2. Performance of 5 Machine Learning Models

As shown in Figure 1, the **Random Forest** and **Multi-layer Perceptron (MLP)** models were the top performers, both achieving an accuracy of **97.9%**. Both models achieved the precision (1.00) on the 'fake' class, meaning it produced zero false positives, meaning they correctly identified every single deepfake sample.

The ensemble models, **XGBoost** and **XGBRF** also demonstrated strong performance, forming a clear second tier. In contrast, kernel-based model of **SVM**, struggled to effectively separate the classes. This wide variance in performance strongly suggests that the relationships within the acoustic feature set are highly non-linear, favoring more complex, ensemble-based, or neural network models that can capture these intricate patterns.

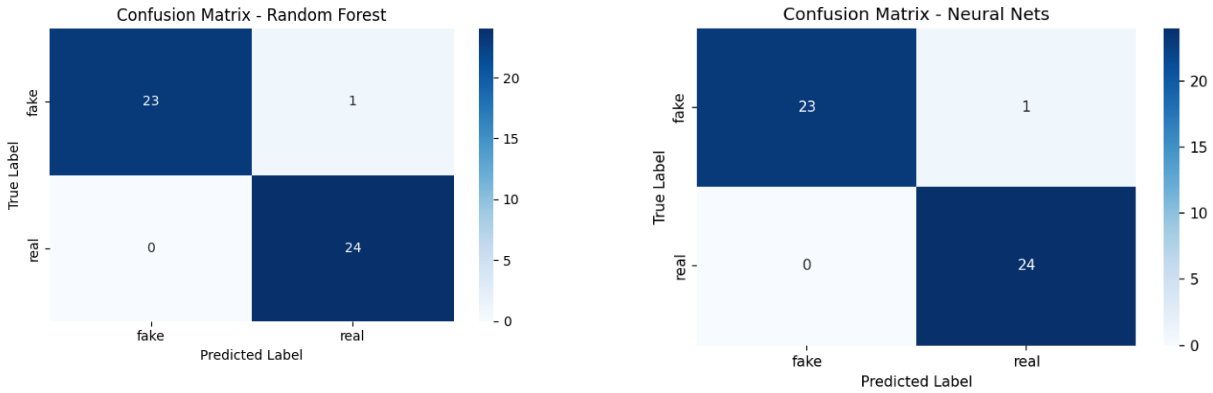

Fig. 3 Confusion Metrics of Random Forest and Neural nets Models

4.3 Testing Results of Deep Learning Models (CNN Models)

The performance of all evaluated models on the respective test sets is summarized in the table below. The deep learning models were trained for 45 epochs, with experiments conducted for both **batch size 16** and **batch size 32** to assess the impact of this hyperparameter.

	Batch Size - 16					Batch Size - 32				
Models	**Average Accuracy**	**Precision**		**F1-score**		**Accuracy of F-1 Score**	**Precision**		**F1-score**	
		Real	**Fake**	**Real**	**Fake**		**Real**	**Fake**	**Real**	**Fake**
ResNet50	93%	96%	90%	93%	93%	84%	86%	84%	80%	87%
ResNet18	91%	97%	86%	92%	91%	84%	100%	76%	82%	87%
ResNet152	93%	91%	96%	94%	92%	83%	90%	78%	79%	85%
ResNet101	93%	93%	93%	93%	93%	93%	93%	93%	93%	93%
AlexNet	97%	93%	100%	96%	97%	97%	93%	100%	96%	97%
MobileNet v2	91%	92%	91%	90%	93%	90%	91%	89%	88%	91%

Fig 4 Performance of CNN Models

The results confirm that the deep learning approaches outperform the classical methods. Notably, **AlexNet trained with a batch size of 16** achieved the highest overall accuracy. For all models, the smaller batch size yielded slightly better results. However, AlexNet's performance stays consistent with the larger batch size of 32, highlighting the nuanced impact of this hyperparameter across different model architectures.

Batch: 16 **Batch:32**

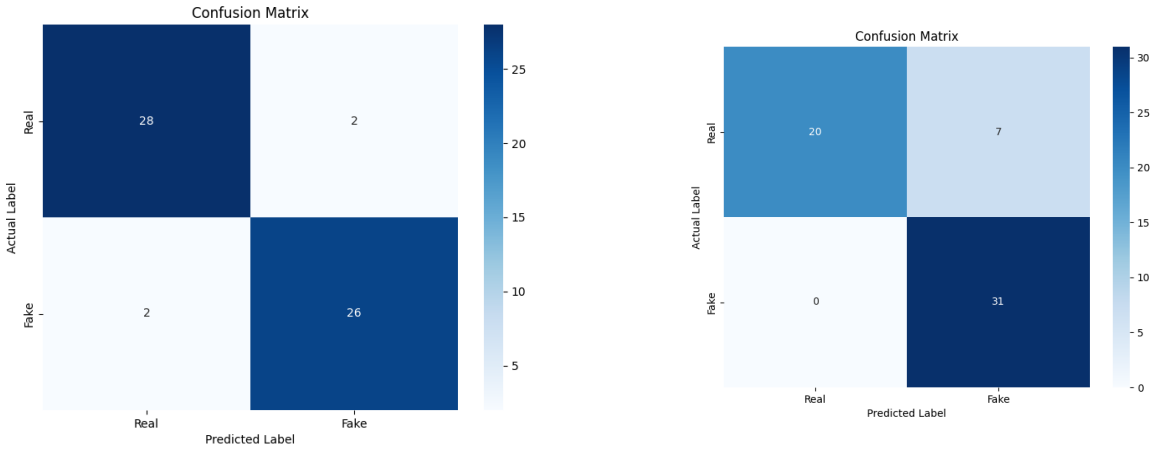

Fig 5. Confusion Matrices of AlexNet Model

4.4 Analysis of Model Performance Across Methodologies

The most striking outcome of the classical model evaluation was the clear performance gap between non-linear models and their counterparts. The top-tier models, Random Forest and the MLP Classifier, both achieved 97.9% accuracy, while the kernel-based SVM struggled with 87.5% accuracy. This suggests that the acoustic relationship between authentic and deepfake audio is not simple or linearly separable. Instead, the "fingerprints" of synthetic speech are embedded in complex, non-linear interactions among the extracted acoustic features, which the more sophisticated models were adept at capturing. The success of these top-performing models is also a direct testament to the efficacy of the feature engineering process, which transformed the complex audio analysis into a structured classification task.

Beyond the performance of individual models, a central finding of this research is that the best classical models outperformed the deep learning (CNN) models on this task. This outcome, while potentially counterintuitive, is explained by the interplay between model complexity and dataset size. The CNN models, like AlexNet, were tasked with the difficult challenge of learning features directly from spectrogram images. Deep learning models are "data-hungry," and the training set size in this study (~134 images) was likely insufficient for them to learn a truly robust representation, even with transfer learning. In contrast, the classical models were given a significant advantage with a curated set of highly informative features. This highlights a key principle: on a moderately sized dataset, a well-engineered feature set can enable a less complex model to achieve superior results.

Building on the performance analysis, the results support the feasibility of the project's main goal: to effectively filter deepfake commands from voice assistant interactions. The fact that most algorithms surpassed **90% accuracy** confirm that a reliable detection mechanism is an achievable objective.

5. Limitations & Future Work

5.1 Limitations

While this study yielded strong results, it is important to acknowledge its limitations and identify avenues for future research.

- **Generalizability to New Generators:** The models were trained and tested on deepfakes from a specific set of four TTS engines. The high accuracy may be due to consistent, learnable artifacts produced by these particular generators. The models' performance is not guaranteed against newer, more advanced deepfake techniques.
- **Real-World Conditions:** The audio data was collected in a **quiet office environment**. The models' performance in real-world scenarios with background noise, microphone variability, and room reverberation was not assessed.
- **Hyperparameter Optimization:** The models in this study were trained with a standard set of hyperparameters. A more exhaustive tuning process could potentially yield further performance improvements, particularly for the CNNs and underperforming models like SVM.

5.2 Future Works

The limitations of this study directly inform the roadmap for future research. The following steps are essential to building upon the current work:

- **Continual Dataset Expansion:** To address the issue of generalizability, a key future task is to continually expand the training dataset with samples from the latest state-of-the-art

deepfake generators. This would involve creating a dynamic pipeline to ensure the detection models remain robust against the rapidly evolving landscape of generative AI.

- **Testing in Realistic Environments:** To ensure real-world viability, future research must involve testing the models on data that has been augmented with various noise profiles (e.g., public spaces, vehicle interiors) and audio compressions to simulate realistic conditions.
- **Exhaustive Hyperparameter Tuning:** A systematic tuning process, using methods like GridSearchCV or RandomizedSearchCV, should be conducted. This would serve to maximize the predictive performance of the models and potentially discover configurations that yield even higher accuracy.
- **System Integration and Prototyping:** A long-term goal is to extend this research by integrating the most effective detection model into a fully functional prototype voice assistant. This would involve developing a complete software pipeline that intercepts an incoming audio command, passes it through the detection model in real-time, and either executes the command if classified as authentic or blocks it if flagged as a deepfake. This step would serve to validate the practical applicability and real-world performance of the filtration technique.

Conclusion

This project addressed the emerging security challenge of deepfake audio targeting voice assistant systems. The primary objective was to develop and evaluate a robust machine learning framework capable of accurately distinguishing between authentic human commands and synthetically generated ones.

Through a comprehensive comparative analysis, this study evaluated two distinct methodologies: a classical machine learning approach using a curated set of acoustic features, and a deep learning approach using mel spectrogram-based image classification. The findings clearly demonstrated that, for a dataset of this scale, the classical approach was more effective. The **Random Forest Classifier** and **Multi-layer Perceptron (MLP) Classifier** emerged as the top-performing models, both achieving an accuracy of **97.9%** on the unseen test data. These models significantly outperformed the Convolutional Neural Network (CNN) architectures.

The success of the classical models underscores the decisive role of **high-quality feature engineering**. By transforming the audio signals into an information-rich feature set, the classification task was made more manageable and effective for models like Random Forest and MLP. This study highlights a critical trade-off in machine learning: while deep learning offers powerful end-to-end capabilities, its performance is highly dependent on large volumes of data. On a moderately sized dataset, a well-designed feature set can enable fewer complex models to achieve superior results.

In conclusion, this research successfully demonstrates that deepfake voice commands can be detected with a high degree of accuracy, presenting a promising proof-of-concept for enhancing the security of voice-activated systems. It confirms that by analyzing fundamental acoustic

properties, it is possible to create an effective detection methodology against current audio synthesis techniques. Future work should build directly on the limitations of this study, focusing on three key areas: 1) expanding the dataset with more diverse and advanced deepfake examples to improve generalizability; 2) evaluating model performance under realistic noise conditions to ensure real-world robustness; and 3) conducting exhaustive hyperparameter optimization to further refine model accuracy.

References

[1] AlSobeh, Anas, et al. "Unmasking Media Illusion: Analytical Survey of Deepfake Video Detection and Emotional Insights." *Issues in Information Systems*, vol. 25, no. 2, 2024, pp. 96-112.

[2] "ASVspoof 2021: Towards Spoofed and Deepfake Speech Detection in the Wild." *Kaggle*, 2022, www.kaggle.com/datasets/mohammedabdeldayem/avsspoof-2021. Accessed 14 June 2025.

[3] Carlini, Nicholas, et al. "Hidden Voice Commands." *Proceedings of the 25th USENIX Security Symposium*, USENIX Association, 2016, pp. 513-30.

[4] Chandra, Nuria Alina, et al. "Deepfake-Eval-2024: A Multi-Modal In-the-Wild Benchmark of Deepfakes Circulated in 2024." *arXiv*, 2025, arXiv:2503.02857.

[5] Frank, J., and L. Schönherr. "WaveFake: A Data Set to Facilitate Audio Deepfake Detection." *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, vol. 1, 2021.

[6] Garg, Ashi, et al. "Less is More for Synthetic Speech Detection in the Wild." *arXiv*, 14 Feb. 2025, arXiv:2502.05674v3.

[7] Gu, Hao, et al. "ALLM4ADD: Unlocking the Capabilities of Audio Large Language Models for Audio Deepfake Detection." *arXiv*, 16 May 2025, arXiv:2505.11079v1.

[8] Kameoka, H., et al. "StarGAN-VC: Non-Parallel Many-to-Many Voice Conversion Using Star Generative Adversarial Networks." *2018 IEEE Spoken Language Technology Workshop (SLT)*, IEEE, 2018, pp. 266-73.

[9] Kassis, A., and U. Hengartner. "Breaking Security-Critical Voice Authentication." *2023 IEEE Symposium on Security and Privacy (SP)*, IEEE, 2023, pp. 951-68.

[10] Kinnunen, Tomi, et al. "ASVspoof 2017: The Second Automatic Speaker Verification Spoofing and Countermeasures Challenge." *Proceedings of Interspeech 2017*, ISCA, 2017, pp. 2093-97.

[11] Kong, J., et al. "HiFi-GAN: Generative Adversarial Networks for Efficient and High Fidelity Speech Synthesis." *arXiv*, 2020, arXiv:2010.05646.

[12] Li, Xinfeng, et al. "SafeEar: Content Privacy-Preserving Audio Deepfake Detection." *Proceedings of the 2024 ACM SIGSAC Conference on Computer and Communications Security (CCS '24)*, ACM, 2024.

[13] Müller, Nicolas, et al. "MLAAD: The Multi-Language Audio Anti-Spoofing Dataset." *International Joint Conference on Neural Networks (IJCNN)*, 2024.

[14] Nautsch, Andreas, et al. "ASVspoof 2019: Spoofing Countermeasures for the Detection of Synthesized, Converted and Replayed Speech." *IEEE Transactions on Biometrics, Behavior, and Identity Science*, vol. 3, no. 2, 2021, pp. 252-65.

[15] "Partial Fake Speech Attacks in the Real World Using Deepfake Audio." *Applied Sciences*, vol. 14, no. 6, 2024, p. 2383.

[16] "The Rise of Voice AI Adoption in a Post-Pandemic World." *AIRudder*, 2023, **www.airudder.com/the-rise-of-voice-ai-adoption-in-a-post-pandemic-world/**. Accessed 13 June 2025.

[17] "Securing Voice Authentication in the Deepfake Era." *Cyberint*, 2024, cyberint.com/blog/thought-leadership/securing-voice-authentication-in-the-deepfake-era/. Accessed 15 June 2025.

[18] van den Oord, A., et al. "WaveNet: A Generative Model for Raw Audio." *arXiv*, 2016, arXiv:1609.03499.

[19] "Voice Assistant Market Size And Forecast." *Verified Market Research*, 2024, www.verifiedmarketresearch.com/product/voice-assistant-market/. Accessed 18 June 2025.

[20] "Voice Clones and Audio Deepfakes: The Security Threats Are Real." *IDRND*, 2024, www.idrnd.ai/voice-clones-and-audio-deepfakes-the-security-threats-are-real/ . Accessed 13 June 2025.

[21] "Voice Search Statistics 2025 (Key Highlights)." *DemandSage*, 2025, www.demandsage.com/voice-search-statistics/. Accessed 15 June 2025.

[22] "Vulnerability Issues in Automatic Speaker Verification (ASV) Systems." *ResearchGate*, 2024, www.researchgate.net/publication/378128004_Vulnerability_issues_in_Automatic_Speaker_Verification_ASV_systems . Accessed 12 June 2025.

[23] Wu, Haolin, et al. "CLAD: Robust Audio Deepfake Detection Against Manipulation Attacks with Contrastive Learning." *arXiv*, 24 Apr. 2024, arXiv:2404.15854v1.

[24] Wu, Zhizheng, et al. "Vulnerability Evaluation of Speaker Verification Under Voice Conversion Spoofing: The Effect of Text Constraints." *Proceedings of Interspeech 2013*, ISCA, 2013.

[25] Xia, Qi, et al. "Near-Ultrasound Inaudible Trojan (Nuit): Exploiting Your Speaker to Attack Your Microphone." *Proceedings of the 32nd USENIX Security Symposium*, USENIX Association, 2023.

[26] Yadav, Amit Kumar Singh, et al. "DSVAE: Interpretable Disentangled Representation for Synthetic Speech Detection." *arXiv*, 28 Jul. 2023, arXiv:2304.03323v2.

[27] Yamamoto, R., et al. "Parallel WaveGAN: A Fast Waveform Generation Model Based on Generative Adversarial Networks with Multi-Resolution Spectrogram." *arXiv*, 2020, arXiv:1910.11480.

[28] Yan, Qiben, et al. "SurfingAttack: Interactive Hidden Attack on Voice Assistants Using Ultrasonic Guided Waves." *Proceedings of the Network and Distributed System Security Symposium (NDSS)*, Internet Society, 2020.

[29] Yan, Ziwei, et al. "VoiceWukong: Benchmarking Deepfake Voice Detection." *Proceedings of the 33rd USENIX Security Symposium*, USENIX Association, 2024.

[30] Zhang, Guoming, et al. "DolphinAttack: Inaudible Voice Commands." *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS '17)*, ACM, 2017, pp. 103-15.