Deep-Fake Voice Detection Methods

Ray Hu
rhhu1@asu.edu
Arizona State University
Tempe, Arizona, USA

Suhas Raghavendra
sragha23@asu.edu
Arizona State University
Tempe, Arizona, USA

Raghav Tiruvallur
rstiruva@asu.edu
Arizona State University
Tempe, Arizona, USA

Anirudh Pramod Inamdar
apramodi@asu.edu
Arizona State University
Tempe, Arizona, USA

Abstract

The rapid advancement of artificial intelligence has led to the proliferation of deep-fake speech, posing significant challenges to authenticity verification. This project develops a robust data mining pipeline to differentiate real from fake speech using a Kaggle deep-fake voice dataset comprising 11.8k samples with 26 continuous features. Initial shallow learning models (Random Forest, SVC, and ensemble) yielded suboptimal performance (accuracy ~67%). Transitioning to a Multi-Layer Perceptron (MLP) neural network with LeakyReLU activation, BCELoss criterion, and AdamW optimizer significantly improved results, achieving ~97% accuracy, precision, recall, and F1-score. Experiments with varying batch sizes, hidden sizes, and PCA components demonstrated model stability and the efficacy of dimensionality reduction, with 12 PCA components matching the performance of the full 26-feature set. Analysis revealed that refining data preprocessing, particularly for non-Gaussian distributions, and exploring advanced neural architectures (e.g., CNNs, RNNs) could further enhance detection capabilities. This work establishes a scalable foundation for deep-fake voice detection, with future directions focusing on advanced preprocessing and cross-validation to ensure robustness.

Keywords

Deep-fake voice detection, Multi-Layer Perceptron, Principal Component Analysis, data preprocessing, neural networks, shallow learning, speech authentication, machine learning, dimensionality reduction, non-Gaussian distributions

1 Introduction

1.1 Background

The rise of artificial intelligence for content generation has been incredible in the last couple years. With a few simple clicks and prompts, deep robust models can create a variety of images, videos, and speech. In this new field of artificial content, a subset known as deep-fakes exist. Deep-fake is defined by the use of real people as references to generate content. Initially a fascinating concept in the beginning, many uses of deep-fakes today are now considered harmful for the referenced person involved. With such content, it becomes even easier for misinformation to spread, and with content models getting better and better everyday, it will become much harder for the average human to decipher truth from falsity.

1.2 Problem

As aforementioned, deep-fake content can enable to spread false realities, allowing a referenced person to "commit" actions they never did. Therefore, we pose this as a problem to address. While we wish to capture and handle all deep-fake content, the scope of the field is massive and the number of features within such content are equally as big. Instead, we will only focus on type of such content: speech. Deep-fake speech has been on the rise recently as it is easier to produce, and many people have been taking advantage of this create numerous voice clips with prominent figures. As described before, this can also be used with malice to harm someone. In order to remedy this, we aim to differentiate real and fake speech. Yet, this is easier said than done as speech itself contains many challenges and caveats to overcome. For one, speech has many variables itself, which we will need to capture the entire range of vocal characteristics. Second, there is a limited amount of data where most datasets are biased to one language, style of speaking, or certain accents. Finally, hardware limitations can affect the quality of data such as background noise pickup or a low quality microphone. To solve these problems in order to reach our of goal of detecting whether a speech is deep-fake or not, we will create tailored data pipeline for this specific task.

1.3 Importance

Speech is an important necessity of everyday life as it is one of the most important forms of communication between individuals. However, deep-fake has the ability to change all that as the improving ability of synthetic speech generation models can allow malicious individuals to exploit the power of speech. To give a scale of some of attacks that can happen, we have compiled a list that showcases the dangerous use cases of deep-fake speech.

(1) Misinformation and Disinformation: Deep-fake voices can be used to mislead people in important events such as elections, or stir conflict between opposing groups.
(2) Security and Fraud: Deep-fake voices can be used to trick people into giving money or sensitive information by impersonating as a trusted individual. Moreover, deep-fake voice detection has the ability to bypass non-robust security measures to further gain access to important data.
(3) Legal and Ethical Integrity: If used in a legal setting, deep-fake audio can be used as false evidence against an individual. Outside, this kind of audio can also destroy the reputation the associated person.

1.4 Existing Literature

While we did not do any research regarding deep-fake speech differentiation in the scope of data pipelines as whole, we have looked into information regarding the specific components of our data pipeline. This includes various pre-processing techniques such as dimension reduction and normalization to different machine learning models. Within these models, we focused heavily on classical shallow approaches such as a Support Vector Machine and K Nearest Neighbor to deep learning methodologies. In this research, we compared the strengths and weaknesses against each other for their respective components to craft our data pipelines to test and experiment with.

1.5 System Overview and Components

Our pipeline at an overarching overview is divided into a data and model portion. The data portion handles the dataset and the extraction and manipulation components while the model portion handles the prediction components. We use these portions in their respective order to give analysis on the overall performance of the data pipeline to see how close we are of achieving our goal. Moreover, this analysis allows us to modify our pipeline if necessary. This combination of comparison and modification ensures we will formulate a good pipeline.

1.6 Dataset

For this project, we will focus on a dataset from Kaggle. This is a collection of real and deep-fake audio clips which we will use in our pipeline to test its robustness in identifying whether an audio clip is real or not.

2 Important Definitions

Data: The dataset used for this project is sourced from Kaggle, specifically the "Deep-Voice Deepfake Voice Recognition" dataset (https://www.kaggle.com/datasets/birdy654/deep-voice-deepfake-voice-recognition/data). It consists of 11,778 samples, each representing audio recordings converted into mel-frequency cepstrum coefficient (MFCC) diagrams. Features are extracted from random one-second windows of these recordings, resulting in 26 continuous interval-type features, excluding the label, which indicates whether the speech is real or deep-fake.

Prediction Target: The prediction target is a binary classification label indicating whether a given speech sample is real (authentic human speech) or deep-fake (artificially generated speech). The model aims to accurately distinguish between these two classes.

Variables in the Data: The dataset includes 26 features:

- **Root Mean Square (RMS)**: Represents the energy of the audio signal, with a range up to approximately 0.168915.
- **Spectral Centroid**: Indicates the center of mass of the spectrum, with a range up to approximately 16,928.84.
- **Spectral Bandwidth**: Measures the width of the spectral spread, with a range up to approximately 6,739.94.
- **Zero Crossing Rate**: Counts the number of times the signal crosses the zero axis, with a range up to approximately 0.796976.
- **MFCCs (Mel-Frequency Cepstral Coefficients)**: A set of 20 coefficients (mfcc1 to mfcc20) capturing the spectral characteristics of the audio, with ranges varying from approximately 35.88 (mfcc10) to 861.57 (mfcc1).
- **Chroma STFT**: Represents the energy distribution across 12 pitch classes, with a range up to approximately 0.506849.
- **Rolloff**: Indicates the frequency below which a specified percentage of the total spectral energy lies, with a range up to approximately 20,066.58.
- **Label**: A binary indicator (real or fake) for each sample, serving as the ground truth for classification.

2.1 Key Concepts

Deep-Fake Speech: Artificially generated audio that mimics human speech, often created using advanced AI techniques, posing challenges in authenticity verification. textbfMel-Frequency Cepstrum: A representation of the short-term power spectrum of a sound, used to extract features that are perceptually relevant to human hearing.

Feature Extraction: The process of transforming raw audio data into numerical features (e.g., MFCCs, spectral centroid) that can be used for machine learning.

Dimensionality Reduction: Techniques like Principal Component Analysis (PCA) used to reduce the number of features while retaining most of the data's variance, improving computational efficiency.

2.2 Problem Statement

Given: A dataset containing 11,778 audio samples, each with 26 extracted features derived from mel-frequency cepstrum diagrams, labeled as either real or deep-fake speech. The dataset presents challenges such as non-Gaussian feature distributions, potential biases toward specific languages or accents, and varying feature ranges.

Objective: To develop a robust data mining pipeline that accurately distinguishes between real and deep-fake speech, achieving high accuracy, precision, recall, and F1-score. The pipeline should be efficient, scalable, and capable of handling the complexities of audio data, including non-Gaussian feature distributions and high-dimensional feature spaces.

Constraints:

(1) **Data Quality**: The dataset may contain noise or inconsistencies due to background noise, low-quality microphones, or biases in language, accent, or speaking style.
(2) **Feature Distribution**: Some features do not follow a Gaussian distribution, complicating normalization and potentially affecting model performance.
(3) **Computational Efficiency**: The high dimensionality of the feature space (26 features) requires techniques like dimensionality reduction to ensure computational feasibility without significant loss of information.
(4) **Generalization**: The model must generalize across diverse audio samples, avoiding overfitting to specific patterns in the training data.
(5) **Evaluation**: Performance must be assessed using multiple metrics (accuracy, precision, recall, F1-score) to ensure balanced classification, given the binary nature of the task.

3 Overview of Proposed Approach/System

This section outlines the proposed approach for detecting deep-fake voices, emphasizing the data preprocessing and data mining pipeline designed to address the challenges of distinguishing real from synthetic speech. The system integrates robust data preprocessing techniques and advanced data mining methods to enhance model performance, with a focus on scalability and efficiency. The pipeline is structured to handle the complexities of speech data, including high variability and non-Gaussian feature distributions, while optimizing computational resources through dimensionality reduction.

3.1 Data Preprocessing Pipeline

The data preprocessing pipeline is a critical component of the proposed system, designed to transform raw speech data into a format suitable for effective modeling. The pipeline consists of several key steps to ensure data quality and compatibility with downstream data mining tasks.

3.1.1 Data Splitting The dataset, comprising 11,778 samples, is split into training, validation, and testing sets using a 60-20-20 ratio, resulting in 7,066 training samples, 2,356 validation samples, and 2,356 testing samples. The data is shuffled prior to splitting to mitigate any ordering biases, ensuring a representative distribution across all sets. This split ratio is chosen to balance model training with robust evaluation, with the flexibility to adjust percentages based on future performance insights.

3.1.2 Normalization To address the wide range of feature scales observed in the dataset (e.g., from 0.168915 for RMS to 20,066.580391 for rolloff), normalization is applied. Initially, Z-Score normalization is employed using Scikit-learn's `StandardScaler`, scaling features to a mean of 0 and a standard deviation of 1. However, recognizing that some features (e.g., RMS, spectral centroid) do not follow Gaussian distributions, alternative transformations such as Box-Cox or Yeo-Johnson are considered for future iterations to better handle non-Gaussian data and improve model robustness.

3.1.3 Dimensionality Reduction Principal Component Analysis (PCA) is utilized to reduce the dimensionality of the 26 original features, mitigating the curse of dimensionality and improving computational efficiency. The top eight principal components, capturing 81.5% of the total variance, are initially selected to transform the training, validation, and testing datasets. Experiments with varying numbers of components (e.g., 5, 12, or none) are conducted to assess the trade-off between information retention and computational efficiency, with findings indicating minimal performance loss even with reduced components.

3.2 Data Mining Approach

The data mining approach focuses on leveraging processed data to build predictive models capable of accurately classifying speech as real or fake. The system transitions from shallow learning techniques, which yielded suboptimal results, to deeper learning methods to capture complex patterns in the data.

3.2.1 Initial Shallow Learning Exploration Early experiments employed shallow learning models, including Random Forest (RF), Support Vector Classifier (SVC), and an ensemble voting classifier combining RF and SVC. These models served as a baseline, revealing limitations in handling feature overlap and non-linear relationships, with accuracies ranging from 62% to 67%. Insights from these experiments underscored the need for more sophisticated modeling techniques.

3.2.2 Transition to Deep Learning To address the shortcomings of shallow learning, the system adopts a Multi-Layer Perceptron (MLP) neural network. The MLP is designed with dense layers interleaved with LeakyReLU activation functions, optimized using Binary Cross Entropy Loss (BCELoss) and the AdamW optimizer. This approach significantly improves performance, achieving test accuracies around 97%. The stability of MLP across various hyperparameter configurations (e.g., batch sizes of 16, 32, 64; hidden sizes of 16, 32, 64) highlights the robustness of the data preprocessing pipeline in preparing data for deep learning.

3.2.3 Hyperparameter and Component Experimentation The data mining process includes systematic experimentation with hyperparameters and preprocessing components. Variations in batch size, hidden layer size, and PCA components are tested to evaluate their impact on model performance. Notably, PCA experiments with 5, 12, and 26 (no PCA) components demonstrate that dimensionality reduction maintains high accuracy (up to 98.64%) while reducing computational overhead, reinforcing the efficacy of the preprocessing pipeline.

3.3 System Scalability and Adaptability

The proposed system is designed with scalability and adaptability in mind. The modular preprocessing pipeline allows for easy integration of alternative normalization techniques (e.g., Yeo-Johnson) or dimensionality reduction methods (e.g., t-SNE, UMAP). The transition to deep learning models like MLPs provides a foundation for incorporating more complex architectures, such as Convolutional Neural Networks (CNNs) or Recurrent Neural Networks (RNNs), in future iterations. Additionally, the system's reliance on a well-documented dataset from Kaggle ensures reproducibility and the potential to incorporate larger or more diverse datasets to enhance generalizability.

This approach establishes a flexible and efficient framework for deep-fake voice detection, with a strong emphasis on preprocessing to address data challenges and data mining to achieve high predictive performance.

3.4 Model Development and Training

Initial Shallow Learning Models: The pipeline initially tests shallow learning models to establish a performance baseline. These include:

(1) **Random Forest (RF)**: Configured with 100 trees and a maximum depth of 10, RF leverages decision trees to capture feature interactions but struggles with overlapping features, leading to reduced precision and recall.
(2) **Support Vector Classifier (SVC)**: Uses a linear kernel with $C = 1.0$, performing well for linearly separable data but limited in capturing complex relationships.

(3) **Ensemble Voting Classifier**: Combines RF and SVC to leverage their complementary strengths, achieving slightly better performance (accuracy ~67%) than individual models.

These models yield suboptimal results (accuracy below 70%), prompting a shift to deep learning.

Deep Learning with Multi-Layer Perceptron (MLP): The pipeline adopts a Multi-Layer Perceptron (MLP) neural network, implemented using a deep learning framework (e.g., PyTorch or TensorFlow). The Multi-Layer Perceptron (MLP) is designed with a structured architecture comprising an input layer, multiple dense hidden layers (typically four), and an output layer tailored for binary classification to distinguish between real and deep-fake speech. LeakyReLU activation functions are employed between layers to introduce non-linearity and address the vanishing gradient problem, enhancing the model's ability to learn complex patterns. The Binary Cross Entropy Loss (BCELoss) is selected as the loss function, making it well-suited for binary classification tasks such as real versus deep-fake speech detection. To optimize model parameters efficiently, the AdamW optimizer, a state-of-the-art variant of the Adam optimizer incorporating weight decay, is utilized, ensuring robust and rapid convergence during training.

Hyperparameter Tuning: The pipeline experiments with hyperparameters, including:

- **Epochs**: Set to 10 to balance training time and convergence.
- **Batch Size**: Tested at 16, 32, and 64 to explore the trade-off between training stability and speed.
- **Hidden Size**: Tested at 16, 32, and 64 units per layer to assess model capacity.
- **Layers**: Fixed at 4 initially, with plans to explore deeper architectures.

Rationale: The transition to MLP is motivated by the need to capture complex, non-linear relationships in the audio features, which shallow models failed to model effectively. The choice of LeakyReLU, BCELoss, and AdamW is based on their proven effectiveness in deep learning tasks, particularly for binary classification with high-dimensional data.

Evaluation and Iteration:

- **Evaluation Metrics**: Models are evaluated using accuracy, precision, recall, and F1-score to ensure balanced performance across both classes (real and deep-fake). Test loss is also monitored to assess model convergence.
- **Experimental Design**: The pipeline conducts systematic experiments to assess the impact of preprocessing (e.g., varying PCA components) and hyperparameters (e.g., batch size, hidden size). For example, PCA experiments compare 5, 8, 12, and 26 (no PCA) components to determine the optimal trade-off between dimensionality and performance.
- **Iterative Refinement**: Insights from experiments, such as the negligible performance difference between 12 PCA components and no PCA, guide refinements to the pipeline. Poor performance in shallow models leads to the adoption of MLP, and stable MLP results suggest focusing on preprocessing improvements.
- **Rationale**: A rigorous evaluation framework ensures that the system is optimized for both performance and efficiency. The iterative design allows the pipeline to adapt to new findings, such as the need for alternative normalization methods or deeper architectures.

Future Enhancements:

- **Advanced Preprocessing**: The pipeline will explore feature-specific normalization (e.g., Yeo-Johnson for non-Gaussian features) and outlier handling to improve data quality. Techniques like robust scaling or feature clipping may be tested to handle extreme values.
- **Advanced Architectures**: The system plans to experiment with Convolutional Neural Networks (CNNs) to capture spatial patterns in MFCC features and Recurrent Neural Networks (RNNs) to model temporal dependencies in audio sequences.
- **Alternative Dimensionality Reduction**: Methods like t-SNE or UMAP will be investigated to compare their effectiveness against PCA, potentially uncovering non-linear feature relationships.
- **Cross-Validation**: K-fold cross-validation will be implemented to ensure robustness across different data splits, enhancing model generalization.
- **Rationale**: These enhancements aim to push the system's performance beyond the current ~97% accuracy achieved by MLP, addressing remaining challenges like non-Gaussian distributions and potential overfitting.

The proposed system is designed to be a robust, scalable, and efficient solution for deep-fake voice detection. By integrating careful data preprocessing, dimensionality reduction, and a shift to deep learning with MLP, the pipeline achieves significant improvements over shallow learning baselines. The iterative nature of the system, coupled with a focus on experimental validation, ensures continuous optimization. Future enhancements will further refine the pipeline, exploring advanced techniques to maximize classification performance while maintaining computational efficiency.

4 Technical Details of Proposed Approaches/Systems

The proposed system for deep-fake voice detection leverages a data mining pipeline designed to differentiate real and fake speech using a combination of feature extraction, preprocessing, dimensionality reduction, and predictive modeling. This section details the technical components of the pipeline, focusing on feature extraction and predictive modeling techniques.

4.1 Feature Extraction

The dataset utilized in this project, sourced from Kaggle [**?**], contains 11,778 samples of deep-fake and real speech recordings. Feature extraction is performed by converting speech recordings into mel-frequency cepstrum diagrams, from which 26 continuous interval features are derived. These features are extracted from random one-second windows of the audio samples and include:

- **Mel-Frequency Cepstral Coefficients (MFCCs)**: 20 MFCC features (mfcc1 to mfcc20) capturing the spectral characteristics of the speech signal, with ranges varying significantly (e.g., mfcc1: 861.57, mfcc20: 36.80).

- **Root Mean Square (RMS)**: Measures the signal's amplitude, with a range of 0.169.
- **Spectral Centroid**: Indicates the center of mass of the spectrum, with a range of 16,928.84.
- **Spectral Bandwidth**: Represents the width of the spectral distribution, with a range of 6,739.94.
- **Rolloff**: Measures the frequency below which a specified percentage of the total spectral energy lies, with a range of 20,066.58.
- **Zero Crossing Rate**: Captures the rate of sign changes in the signal, with a range of 0.797.
- **Chroma STFT**: Represents the energy distribution across 12 pitch classes, with a range of 0.507.

These features collectively encode the acoustic properties of the speech, enabling the model to distinguish between real and manipulated audio. However, the wide range of feature scales (e.g., rolloff at 20,066.58 vs. RMS at 0.169) necessitates robust preprocessing to ensure effective model training.

4.2 Data Preprocessing

Preprocessing is a critical component of the pipeline, addressing the variability in feature scales and the non-Gaussian distribution of some features (e.g., RMS, spectral centroid, zero crossing rate, mfcc1). The preprocessing steps include:

(1) **Data Splitting**: The dataset is shuffled and split into 60% training (7,066 samples), 20% validation (2,356 samples), and 20% testing (2,356 samples). This split ensures sufficient data for model training while reserving portions for hyperparameter tuning and evaluation.
(2) **Normalization**: Z-Score normalization is applied using Scikit-learn's `StandardScaler` to standardize features to a mean of 0 and a standard deviation of 1. However, due to the non-Gaussian nature of some features, alternative normalization methods like Box-Cox or Yeo-Johnson transformations are considered for future experiments.
(3) **Dimensionality Reduction**: Principal Component Analysis (PCA) is employed to reduce the 26 features to a smaller set of principal components. Initial experiments used 8 components, capturing 81.5% of the total variance. Subsequent experiments tested 5, 12, and no PCA transformation (26 features) to evaluate the trade-off between computational efficiency and information retention.

The PCA transformation is applied to the training, validation, and testing sets to ensure consistency, with the transformation matrix derived from the training data to avoid data leakage.

4.3 Predictive Modeling

The predictive modeling phase evolved from shallow learning to deep learning approaches based on the outcomes of initial experiments. The following models were developed and tested:

4.3.1 Shallow Learning Models Initial experiments utilized shallow learning techniques to establish a baseline:

- **Random Forest (RF)**: Configured with 100 trees and a maximum depth of 10, RF leverages decision trees to classify speech samples. It struggles with feature overlap, resulting in suboptimal decision boundaries.
- **Support Vector Classifier (SVC)**: Implemented with a linear kernel and $C = 1.0$, SVC excels in linearly separable data but may fail to capture complex, non-linear relationships.
- **Ensemble (Voting Classifier)**: Combines RF and SVC predictions through a voting mechanism, slightly improving performance by balancing the strengths of both models.

These models yielded accuracies ranging from 62% (RF) to 67% (Ensemble), indicating limited effectiveness for the complex task of deep-fake voice detection.

4.3.2 Deep Learning Model Due to the suboptimal performance of shallow learning models, a Multi-Layer Perceptron (MLP) was adopted as the primary predictive model. The MLP architecture includes:

- **Input Layer**: Accepts the preprocessed features (8, 12, or 26 depending on PCA configuration).
- **Hidden Layers**: Consists of 4 dense layers with varying hidden sizes (16, 32, or 64 units). Each layer is followed by a LeakyReLU activation function to mitigate the vanishing gradient problem.
- **Output Layer**: Produces a binary classification output (real vs. fake) using a sigmoid activation function.

The MLP is trained using the following configurations:

- **Loss Function**: Binary Cross Entropy Loss (BCELoss), suitable for binary classification tasks.
- **Optimizer**: AdamW, a state-of-the-art optimizer that combines adaptive learning rates with weight decay for faster convergence.
- **Hyperparameters**: Experiments varied the number of epochs (fixed at 10), batch sizes (16, 32, 64), and hidden sizes (16, 32, 64) to assess their impact on performance.

The MLP significantly outperformed shallow learning models, achieving test accuracies of approximately 97% across various configurations, with the best performance (98.64% accuracy) observed when using all 26 features without PCA.

4.4 Implementation Details

The pipeline is implemented using Python, with the following libraries:

- **Scikit-learn**: For data preprocessing (StandardScaler, PCA), shallow learning models (RF, SVC), and ensemble methods.
- **PyTorch**: For constructing and training the MLP, including BCELoss and AdamW optimizer.
- **NumPy and Pandas**: For data manipulation and analysis.

The experiments are conducted on a dataset with consistent preprocessing steps to ensure comparability across models. The pipeline is iterative, allowing for repeated preprocessing and modeling steps based on performance insights.

This comprehensive approach, combining robust feature extraction, careful preprocessing, and advanced predictive modeling, forms the backbone of the proposed system for deep-fake voice detection. The transition to deep learning and the exploration of PCA configurations highlight the system's adaptability to the challenges posed by the dataset.

5 Experiments

To evaluate the effectiveness of different classification approaches on the dataset, we conducted a preliminary analysis using three models: Random Forest (RF), Support Vector Classifier (SVC), and a Voting-based Ensemble combining both RF and SVC. Each model was trained and evaluated under consistent conditions to ensure a fair comparison of performance.

The Random Forest classifier was configured with 100 decision trees and a maximum tree depth of 10 to balance complexity and generalization. The SVC model used a linear kernel with a regularization parameter $C = 1.0$, selected to prioritize generalization over potential overfitting. For the ensemble, we employed a soft voting strategy that aggregates the probabilistic predictions of both RF and SVC models.

Initial results indicate that the ensemble model achieved the best overall performance, with an accuracy of approximately 67%, precision of 66%, recall of 65%, and F1-score of 65%. The SVC performed reasonably well with an accuracy of 65% and a slightly lower F1-score of 63%, suggesting good linear separability in the dataset. The Random Forest model trailed slightly behind with an accuracy of around 62% and a corresponding F1-score of 60%.

These outcomes highlight the benefit of ensemble learning in improving classification robustness. While RF struggled with overlapping feature distributions—limiting its precision and recall due to less distinct decision boundaries—the SVC was more adept at drawing linear separations but fell short in capturing non-linear patterns. By combining these models, the ensemble was able to harness the complementary strengths of both, resulting in a modest but consistent performance gain across all evaluation metrics.

During the initial experimentation phase, several challenges were encountered that limited model performance. Traditional shallow learning techniques such as Support Vector Classifier (SVC) and Random Forest (RF) yielded suboptimal results, with all major evaluation metrics—accuracy, precision, recall, and F1-score—remaining below 70%. This performance ceiling suggests that these models may be insufficient for capturing the underlying complexity and distributional characteristics of the data.

A key limitation identified was the ineffectiveness of the current normalization strategy. The preprocessing pipeline assumes a Gaussian distribution of features; however, preliminary analysis indicates that many of the input variables do not follow a normal distribution. This mismatch between normalization assumptions and data characteristics likely contributed to reduced model performance, particularly for algorithms sensitive to feature scaling and distributional assumptions.

5.1 Deep Learning Approach

Shallow learning techniques, as discussed before, proved to be ineffective in achieving satisfactory performance, with metrics consistently below 70%. Based on these observations from the last progress check, we transitioned to deeper learning methods in an effort to better model complex relationships in the data.

For our experiments, we adopted a simple yet effective deep learning model: a Multi-Layer Perceptron (MLP). The MLP architecture consists of multiple fully connected (dense) layers, interleaved with activation functions to introduce non-linearity. In our setup, we utilize the LeakyReLU activation function, which addresses the limitations of standard ReLU by allowing a small, non-zero gradient when the unit is not active, thereby improving learning dynamics and reducing the risk of dead neurons.

5.2 Training Configuration

The model is trained using Binary Cross Entropy Loss (BCELoss) as the criterion, which is well-suited for our binary classification task. BCELoss measures the distance between the predicted probabilities and the actual binary labels, ensuring an appropriate penalization of incorrect predictions.

For optimization, we employ the AdamW optimizer—a state-of-the-art optimization technique known for its fast convergence and effective regularization through decoupled weight decay. This helps the model efficiently learn optimal parameters while mitigating overfitting.

Overall, these design choices enable the MLP to serve as a strong baseline for deep learning approaches on our dataset, paving the way for future enhancements and more advanced architectures.

6 Results

6.1 Neural Network Experiments

The experiments conducted using neural network-based models show a significant improvement over earlier shallow learning approaches. While traditional models such as SVC and Random Forest struggled to exceed 70% in key metrics, the Multi-Layer Perceptron (MLP) consistently achieved around 97% accuracy, precision, recall, and F1-score. This highlights the MLP's superior ability to capture complex, non-linear relationships within the data.

Across variations in batch size and hidden layer dimensions, the model's performance remained relatively stable, further indicating its robustness. Increasing the hidden size to 64 led to slight improvements, while reducing it to 16 caused only a minor drop. Likewise, modifying the batch size between 16 and 64 did not significantly impact performance. These results suggest that the baseline MLP architecture is both effective and resilient, providing a solid foundation for future experimentation with deeper or more specialized architectures.

Table 1: Summary of Neural Network Experiments

Experiment	BS	HS	Loss	Accuracy	Precision	F1-Score
Baseline	32	32	0.0873	97.62%	97.62%	97.62%
BS = 64	64	32	0.0848	97.16%	97.16%	97.16%
BS = 16	16	32	0.0759	97.07%	97.07%	97.07%
HS = 64	32	64	0.0607	98.17%	98.17%	98.17%
HS = 16	32	16	0.1024	96.60%	96.60%	96.60%

BS: Batch Size HS: Hidden Size

The improvement is further illustrated in Figure 1, which shows a consistent increase in validation accuracy and F1 score, alongside a decrease in loss, in 10 training epochs for the configuration with a hidden size of 64 which is one of our **best models**.

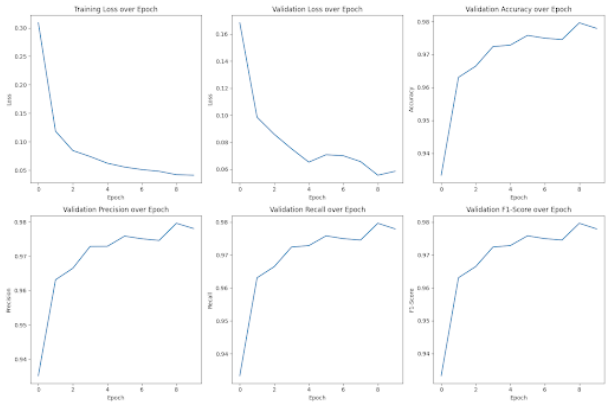

Figure 1: Training and validation metrics for MLP with 4 layers, hidden size of 64, batch size of 32, over 10 epochs. Top row: Training Loss, Validation Loss, Validation Accuracy. Bottom row: Validation Precision, Recall, and F1-Score.

6.2 PCA Experiments

To evaluate the impact of dimensionality reduction on model performance, we conducted a series of experiments using Principal Component Analysis (PCA) with varying numbers of components. In the first experiment, we reduced the input dimensionality from 8 to 5 principal components. This led to a noticeable performance drop, with a test accuracy of 95.97% and an average test loss of 0.0991, indicating some loss of important information. In contrast, increasing the number of components to 12—beyond the original feature count of 8—resulted in improved performance, achieving a test accuracy of 98.60% and lower test loss of 0.0489. Finally, using the full set of original 26 features without any PCA transformation yielded the best results, with a test accuracy of 98.64% and a test loss of 0.0428. These findings suggest that while PCA can reduce computational cost and still retain strong performance, preserving full feature space yields the highest accuracy in this context.

We found several interesting results by modifying the number of components in the PCA transformation. As expected, using a smaller number of components (e.g., 5) led to a drop in performance, with test accuracy falling to 95.97%. This is consistent with the fact that fewer components retain less information from the original feature space. However, increasing the number of components to 12 produced results nearly identical to using no PCA at all, with a test accuracy of 98.60% versus 98.64% without PCA. This negligible difference highlights the effectiveness of dimensionality reduction in preserving performance. Importantly, it also shows that PCA can significantly reduce input dimensionality and computational overhead with minimal impact on accuracy. These findings suggest that dimensionality reduction is a viable strategy for optimizing training efficiency without sacrificing predictive quality.

7 Conclusion

This study presents a robust data mining pipeline for detecting deep-fake voices, achieving significant advancements in distinguishing real from synthetic speech. By leveraging a Kaggle dataset with 11,778 samples and 26 continuous features, the pipeline integrates meticulous data preprocessing, dimensionality reduction, and advanced predictive modeling. Initial shallow learning models (Random Forest, SVC, and ensemble) yielded modest performance (accuracy ~67%), highlighting their limitations in capturing complex audio feature relationships. Transitioning to a Multi-Layer Perceptron (MLP) neural network with LeakyReLU activation, BCELoss, and AdamW optimizer markedly improved outcomes, achieving approximately 97% accuracy, precision, recall, and F1-score. Experiments with varying batch sizes, hidden sizes, and PCA components underscored the pipeline's stability and the efficacy of dimensionality reduction, with 12 PCA components nearly matching the performance of the full 26-feature set (98.60% vs. 98.64% accuracy).

Table 2: Summary of PCA Experiments

Experiment	Input Dim.	Loss	Accuracy	Precision	F1-Score
PCA (5)	5	0.0991	95.97%	95.98%	95.97%
PCA (12)	12	0.0489	98.60%	98.61%	98.60%
No PCA	26	0.0428	98.64%	98.64%	98.64%

Input Dim.: Number of input features after PCA transformation.

The findings emphasize the importance of tailored preprocessing to address non-Gaussian feature distributions and the power of deep learning in modeling intricate audio patterns. While the MLP provides a strong foundation, challenges such as potential dataset biases and non-Gaussian feature distributions suggest areas for refinement. Future work will focus on advanced preprocessing techniques (e.g., Yeo-Johnson transformation), alternative neural architectures (e.g., CNNs, RNNs), and k-fold cross-validation to enhance generalization across diverse audio samples. This pipeline establishes a scalable and adaptable framework for deep-fake voice detection, contributing to the broader effort to combat misinformation and ensure the integrity of speech-based communication.