ADROID Conference Series: Engineering and Technology
Volume 2, Issue 2, pp.55-67, 2026

Deepfake Audio Detection Using MFCC-Based Acoustic Feature Extraction and Machine Learning Classification: A Comprehensive Framework

Soniya Motiwal*, Balaji Patil
Department of Computer Engineering and Technology, Dr. Vishwanath Karad MIT World Peace University, Pune, India
Corresponding author: soniya.motiwal@mitwpu.edu.in*, balaji.patil@mitwpu.edu.in

Abstract. Voice has always been one of the more trusted ways to identify people; we can identify our colleagues, family, and public figures by their voices. Meanwhile, that trust is being systematically exploited. The state of neural speech synthesis today means that AI-generated speech can sound convincing in casual conversations and, as in a handful of cited cases, has been used for financial fraud and targeted deception. In this paper, a detection system using Mel-Frequency Cepstral Coefficients (MFCCs) and a Support Vector Machine (SVM) classifier is presented to classify audio clips as real or generated. The system does not only use the direct MFCC values, but it also computes five other complementary spectral features: spectral centroid, spectral roll-off, zero crossing rate, RMS energy and spectral bandwidth, from which it forms a 45-dimensional feature vector. Noise is filtered, loudness normalised, duration trimmed, and silence standardised via audio pre-processing before any features are computed. The trained classifier is encapsulated within a Flask web app that accepts uploaded audio files, performs inference, and outputs predictions in real time. Results on a balanced benchmark test set of 4,000 clips yielded 95.2% accuracy, 94.1% precision, 95.8% recall, and 94.9% F1-score. These numbers outperform several CNN and TCN baselines in the recent literature at a significantly reduced computational cost. The paper suggests that when executed conscientiously, principled feature engineering can be a viable alternative to large-scale deep learning for this class of problems.

Keywords: Deepfake audio detection, MFCC, SVM, audio forensics, speech processing, machine learning, cybersecurity, and synthetic speech.

1. INTRODUCTION

A time, at least, when voice recording had something of an evidentiary value that could not be easily disputed. Audio evidence was accepted in court, banks used voice prompts to verify their customer's identities, and most people believed that the voice at the other end of a telephone call was what it said it was. This assumption has been eroded over time. The research that led to the impressive advances in speech synthesis has also provided an impersonation tool to anyone with an Internet connection and a modest amount of technical expertise as collateral.

Today, with the advent of more recent text-to-speech systems like WaveNet [22], Tacotron 2 [23] and Deep Voice 3 [26] and others, speech can be perceptually convincing, when only a short amount of reference speech from a target speaker is provided. The Wall Street Journal 2019 report [8] about an incident in which attackers were able to spoof a CEO using AI-generated audio and direct a fake bank transfer was a prime example of how technology moves from research labs to criminal use. More recently, there have been reports of fake political speeches, fake emergency calls that have been used to obtain funds from family members and fake recordings in court cases. Although human recognition of synthetic speech has not kept up with the realism of the speech synthesis, automated detection is a practical necessity and not an academic pursuit [9].

The challenge is technical, and it's very real. The samples are synthesized by various architectures, which lead to various artefacts, and the latest neural vocoders leave increasingly weak artefacts behind [14] [32]. A useful detection system should be able to detect features that are discriminative between the various synthesis methods, recording environments and speaker sets—not just features that work well within a carefully constructed dataset. This requirement led to a number of the choices made in the design described in this paper.

Motivation

However, this field of research is prone to copycat thinking and recent transformer-based systems have impressive benchmark scores [2], [33]. However, there are other factors at play in real deployment situations besides raw accuracy. A forensic tool designed to mark audio evidence in a legal case should be interpretable, that is, it should be possible to explain, in simple terms, why a particular result was reached by the system. A Fraud detection module in the telephony platform must be fast, hardware agnostic, and without the need for GPU infrastructure. A tool used in a multilingual customer support environment must be general enough to perform reasonable unification across languages without the need to start again for new dialects.

Mahendrakumar Kalal et al. (eds.), Proceedings of ICNCDA -2026, Adroid Conference Series: Engineering and Technology. ISBN: 978-81-995593-4-9

Some of these practical restrictions can be overcome by MFCC based feature extraction. They are based on the theory of auditory perception and thus have a physical interpretation, which the activation of the models of deep networks does not so easily provide. They can be computed without a lot of effort. There exists a robust set of evidence that they are sensitive to the spectral irregularities which synthesis pipelines are likely to generate [1], [30]. In this paper, we investigate the feasibility of using an MFCC-plus-spectral system for classification against deep learning alternatives while still maintaining these practical benefits.

Main Contributions
The work described in this paper makes the following contributions:

An end-to-end pipeline for deepfake audio detection, which integrates 40 MFCC coefficients and five spectral descriptors into a 45-dimensional feature representation, where all the decisions made during the pipeline from the pre-processing to the features extraction have been explicitly documented.
Systematic evaluation on a balanced benchmark dataset through stratified train/validation/test splits, accuracy/precision/recall/ F1-score and confusion matrix.
A modular Flask web application that receives audio uploads and provides real-time predictions, highlighting that it is a system that is not only a lab demonstration, but is also practically deployable.
A comparative analysis that compares the proposed approach with the other existing approaches in the literature namely MFCC-based, CNN-based and TCN-based.
An honest assessment of its weaknesses, specifically in noisy environments, for multilingual speech and against adaptive adversaries, and what would be required to overcome each of these.

Paper Organisation
In the second section the previous research is examined. The proposed methodology described in Section 3 consists of data set, pre-processing, feature extraction, normalisation and classification. The architecture and implementation of the system are described in Section 4. The experimental results are presented and their implications are discussed in Section 5. The challenges of the present method are discussed in Section 6. Section 7 provides directions for future development. The paper ends in section 8.

2. LITERATURE REVIEW

So far, the most common application of audio spoofing has been related to speaker verification systems and the question is whether or not an audio recording of an enrolled speaker can be replayed to fool a voice authentication system. The ASVspoof challenge series formalised this threat model and provided the research community with a common benchmark. ASVspoof 2015 [11] concentrated on text-to-speech and voice conversion attacks, showing that, under controlled conditions, many of the synthesis systems used at the time could be correctly identified with low equal error rates. The catch is that generalization to attack classes outside of those encountered in training was significantly poorer — a cautionary note about generalization to particular synthesis artifacts that is relevant today.

However, replay attacks, where an audio sample of a genuine speaker is played back, were the focus of ASVspoof 2017 [12] and of the following day's Verification Challenge. This was a much more difficult task to achieve due to the environmental and channel variability, and methods which relied on features in the frequency domain, and especially on cosine-based features, were more successful than those based on features in the time domain. Text-to-speech attacks and voice conversion attacks were also reintroduced with ASVspoof 2019 [13] and the combined framework for evaluating combined attacks on text-to-speech and automatic speaker recognition, which is still the basis for most current benchmarking, was added. The three challenges influenced the vocabulary and convention for evaluating in the field.

The end-to-end neural synthesis was what made the detection problem fundamentally different. Prior to WaveNet [22] text-to-speech systems used units of recorded speech or a statistical parametric approach that produced characteristic artefacts which could be reasonably well recognised by a good classifier. WaveNet created audio sample by sample using dilated causal convolutions and generated speech that was qualitatively different, more natural, and less prone to the 'metal' and 'buzz' artifacts of prior synthetic speech. In Tacotron 2 [23] they combined this vocoder with a sequence-to-sequence acoustic model, allowing for the creation of convincing speech directly from text without any engineering of intermediate representations.

If the result of detection were just to have classifiers trained on artefacts created by the previous synthesis systems, then the classifiers would not perform well on the newer systems. However, Frank and Schönherr [24] have developed the WaveFake database specifically with the aim of providing a benchmark against neural vocoder results, and found that many systems developed with the earlier style of synthesis had significant performance deficits. This generalisation problem, which is to perform good on one synthesis method and bad on another, has been one of the major problems in the detection task [14, 32].

MFCCs are the most tested classical feature engineering representation. Their architecture is similar to the frequency resolution of the human cochlea, and they have theoretical foundations as a representation of perceptually relevant spectral information. Reimao and Tzerpos [25] proposed the FoR (Fake or Real) dataset and tested several classifiers, namely SVM, Naive Bayes, decision tree and random forest with timbre-based descriptors together with variants of MFCCs. The highest accuracy they achieved using an SVM was 73.46%, thus providing a good lower bound reference to which future work could be compared. Khochare et al. [30] used the same FoR corpus, but with 20 MFCC coefficients, and achieved 67% accuracy with SVM — again, this shows how much performance relies on decisions made prior to the presentation of data to the SVM, as the two SVM systems vary considerably despite implementing the same algorithm.

Almutairi and Elgibreen [14] performed a detailed study of the audio deepfake detection methods and noted an important point: the observed differences in performance across the studies tend to be larger than the differences among the classifiers used, and are generally explained by the particular attack type used and the pre-processing pipeline applied. This discovery was one of the reasons that the current work was very careful to ensure that it had a detailed preprocessing phase as opposed to standardisation as an afterthought.

Over deep learning methods, Wijetunga et al. [7] used CNN and VGG19 network architectures on per-frame MFCC representations and achieved around 89% accuracy on their test set. Khochare et al. [30] obtained a 92% result with TCN architecture that takes mel-spectrogram as input. Verma et al. [2] evaluated CNN, TCN, and transformer models on various datasets and found their accuracy to lie between 91–93%, transformer models being the highest. Comparable to the proposed system, the MFCC features were fed to a VGG-16 architecture by Hamza et al. [1] which achieved a 93% accuracy on FoR dataset, particularly important for comparison, because they applied the same input representation used in the proposed system, before deep network takes over.

Abdulhamied et al. [34] tested this same dataset with the ANN-based approach that utilizes the wavelets and achieved a comparable raw accuracy score with the ResNet-50 approach and came to the same conclusion as in several other studies presented here: the power of deep models does not eradicate the need for classical feature-based approaches that provide interpretability. Jiang et al. [33] brought up a dual-branch CNN model incorporating an attention mechanism achieving good performance by selectively focusing on the temporal regions with synthesis artefacts, suggesting a promising direction for hybrid models.

The system presented in this paper was designed with the following patterns in mind. A frequent issue is missing information, for example, studies may not provide a complete representation of how MFCC features are combined across time, the nature of the data sets at the clip level, or details of the hyperparameters used during the classifier's training. It is truly hard to understand what led to the reported results or to try to replicate the results on a different dataset. The other common shortcoming is the use of a relatively small number of coefficients (usually between 13 and 20) and the fact that most of the SVM system proposed in the literature did not consider whether the cepstral representation is sufficient alone or if other spectral features are suitable for complementing it. The proposed system fulfills both these gaps: First, it is completely explicit about all the choices made (experimental) and second, it tests whether or not augmenting the 40 MFCC coefficients with 5 targeted spectral descriptors improve the MFCC-only baseline in a measurable way. The studies most closely related to this work are summarized in Table 1.

TABLE 1: SHOWS PRIOR STUDIES AND THE PROPOSED MFCC-SVM APPROACH

Ref.	Feature Set	Classifier	Dataset / Setting	Reported Acc.
[25]	Timbre + MFCC variants	SVM, NB, DT, RF	FoR	73.46% (SVM)
[30]	20 MFCC coeffs	SVM, RF, KNN, XGB	FoR original	67% (SVM)
[7]	Per-frame MFCC	CNN, VGG19	UrbanSound8K / conv.	~89%
[2]	Mel-spec, CQT, MFCC	CNN, TCN, transformers	Multiple	91–93% DL models
[14]	Survey (mixed)	ML and DL families	ASVspoof, FoR, etc.	Varies by attack
Proposed	40 MFCC + 5 spectral	SVM (RBF)	Benchmark split	95.2%

3. PROPOSED METHODOLOGY

The detection pipeline consists of audio preprocessing, MFCC extraction, spectral feature extraction, feature vector construction, normalisation and SVM classification. These stages are explained below in the order that they run. The entire pipeline is described in Figure 1.

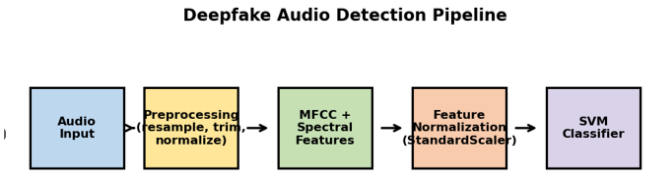

Fig. 1. The entire detection pipeline: raw audio signal from the input, through to preprocessing, feature extraction using MFCC and spectral features, normalisation and classifier (SVM) classification to the binary prediction output.

Dataset

The experiments referenced a balanced benchmark set comprising of 4,000 audio clips, half of which were real and the other half were AI-generated. All clips had 16kHz mono 3-second processing. The authentic recordings were selected from the publicly available speaker corpora that cover a variety of speaker age, gender, regional accent, and recording environments. To avoid the bias towards artefacts characteristic of a particular generation method, the synthetic samples were generated by several different neural synthesis systems, both text-to-speech and voice conversion.

The dataset was equally partitioned into training (70%, validation (15%) and test (15%) sets, with stratified sampling to ensure class proportions were equal across all sets. This resulted in 2,800 training samples, 600 validation samples and 600 test samples. The test split is done separately, before any training or hyperparameter tuning, and every performance is measured on this test partition.

Audio Preprocessing

The recording quality of raw audio files can range from good to bad, as well as background noise, loudness, and length. This variability would cause differences in the samples if left uncompensated, thus making accurate classification more difficult for the classifier to determine when the speech is real or synthetic. Each file is passed through 5 pre-processing steps before features are extracted.

The first operation is resampling, where each file is converted to 16 kHz mono, meaning the overall time-frequency resolution becomes the same for all recordings, no matter what type they originally used. Second, a noise reduction technique based on spectral subtraction is used to estimate and remove the noise component of the recording environment, thus minimizing the impact of the environment on the feature values. Third, since loudness differences may be mistakenly interpreted as discriminative cues, amplitude normalisation ensures that the peak level is fixed in reference. Fourth, silence trimming is performed to remove leading and trailing periods of silence, with an energy threshold, so that the feature extraction window is concentrated on voiced content. Fifth, clips will be padded or shortened to exactly 3 seconds – short clips will be zero padded on the end, and long clips will be cut off at the end.

These five operations are shown in Figure 2.

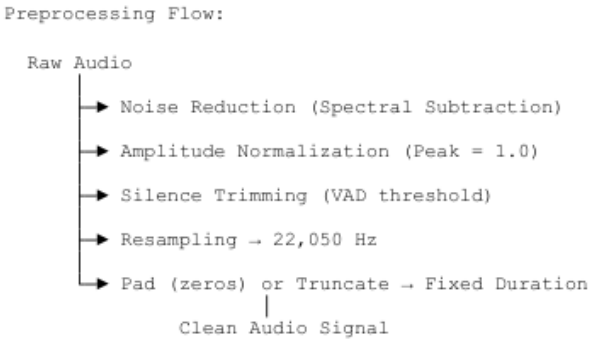

Fig. 2. The different pre-processing operations that were performed on each audio sample before the features were extracted, one after the other from the input raw audio to the standardised output.

MFCC Feature Extraction

The MFCC extraction process is made up of four stages of pipeline. The Short-Time Fourier Transform (STFT) is computed on the preprocessed waveform with a Hamming window of 25 ms and a frame shift of 10 ms to obtain the time-frequency representation of the signal. The squared magnitude spectrogram is then fed into a bank of 40 triangular Mel scale filters. The conversion from linear frequency to the Mel scale is given by:

$$mel(f) = 2595 \cdot log10(1 + 700f)$$

The frequency axis is compressed such that the higher frequencies show less difference in scale than are actually perceived by the auditory system. Log energy is calculated in each filter band using compression so as to simulate the non-linear loudness response of human auditory perception. Finally, the log filter-bank energies are transformed to decorrelated cepstral coefficients using Discrete Cosine Transform. The first 40 co-efficients are kept.

Instead of using the coefficient values per frame the temporal mean of each coefficient across all frames is calculated, which allows for a fixed-length sequence, no padding and no need for a recurrent classifier. This transforms each audio clip into a single 40-dimensional vector, irrespective of any standardisation difference in the remaining duration of the audio clips.

The four-stage extraction pipeline is shown in figure 3

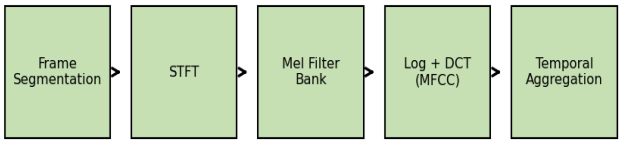

Fig. 3. The pipeline of the MFCC extraction with successive stages from the STFT to the Mel filterbank processing, log compression, and DCT to the 40-coefficient mean vector.

Complementary Spectral Features

The 5 extra descriptors were chosen to cover other aspects of the audio signal that the cepstral coefficients do not adequately represent.

The frequency weighted mean of the power spectrum (informally, the centre of gravity of the spectral energy) is called Spectral Centroid. It matches against perceived brightness and is not always the same for natural voiced speech and flatter or other differently shaped energy distributions as those produced by some synthesis architectures.

The frequency at which 85% of the total spectral energy is contained is called the "spectral roll-off" frequency. It gives an indication of spectral skewness, and is especially useful for identifying voiced speech (which tends to focus energy into lower frequencies) from high energy synthesis artefacts (which tend to leak into higher frequencies).

Zero Crossing Rate (ZCR) is the number of times the waveform changes sign (c crosses the 0 line) over a given time interval. It is very sensitive to voicing characteristics and to the presence of noise-like components, which may vary in genuine as compared to synthetic speech.

Root mean square energy (RMS) is the average power of the signal over the entire length of the signal clip. Some synthesis systems create abnormal uniform energy profiles (i.e. without natural micro-variations in loudness found in human speech) and RMS gives a scalar value of this property.

A measure of the spread of the spectral energy is the Spectral Bandwidth, which is the weighted average standard deviation around the Spectral Centroid of the power spectrum. In areas where there is normally variation in natural speech, narrow bandwidth can be an unobtrusive sign of synthesis.

Then the temporal means of each descriptor are calculated over all of the frames and five scalar values are obtained. These are appended with the 40 MFCC means to form the 45 dimensional feature vector.

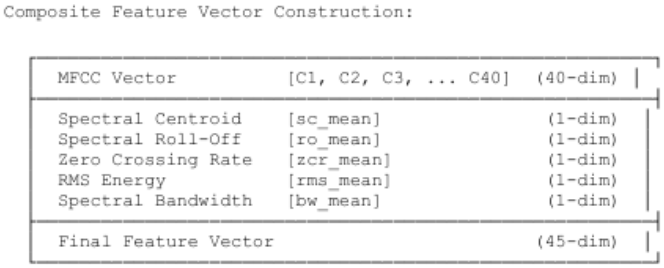

Fig. 4. Construction of the 45-dimensional composite feature vector by concatenating the 40-dimensional MFCC vector with five complementary spectral feature values.

Feature Normalisation

The 45 features span substantially different numerical ranges. MFCC coefficients typically fall between roughly −200 and 200, ZCR values are small positive fractions, and RMS energy values are close to zero for normalised audio. Without standardisation, the SVM's distance calculations in feature space would be dominated by the features with the largest absolute values, regardless of their actual discriminative contribution. StandardScaler normalisation projects each feature to zero mean and unit variance:

$$xscaled = \sigma x - \mu$$

Where μ and σ are computed from the training partition only. The same fitted scaler is applied without modification to the validation and test partitions. Both the scaler and the trained classifier are serialised together for deployment so that inference at test time uses exactly the same transformation that was applied during training.

SVM Classification

The binary classifiers are Support Vector Machines with an RBF (Radial Basis Function) kernel function. SVM learns a maximum margin separation of genuine speech vs. synthetic speech in the normalised feature space, which accounts for the non-linear structure of the class boundaries in the feature space of the feature distributions, through kernel mapping. The decision function is:

$$f(x) = w \cdot \phi(x) + b$$

Where ϕ(x) is the linear combination of the RBF basis functions b is the bias term. The regularisation parameter C and γ the hyperparameters were tuned using the validation partition, grid search and the criterion F1-score. The SVM was chosen over the deep learning alternatives for three important features relevant to the intended deployment: no need for a GPU infrastructure for training or inference; generalization behavior from a medium-sized, balanced set of data, unlike data-hungry deep networks; and decision structure, although not easily interpretable for an arbitrary input, is more easily interpretable in the context of forensic audit than the activations of a large neural classifier.

4. ALGORITHM

Table 2 summarises the MFCC-SVM deepfake audio detection workflow used in this study. The audio signals are then standardised by resampling, removing silence, normalising the amplitude, adjusting the duration, and noise reduction. Finally, MFCC and spectral descriptors are extracted, and fused into a 45-dimensional feature vector. The resulting features are split into training and validation and test sets prior to scaling and training SVM. The hyperparameters are optimised by grid-search based on the F1-score in the validation-set. The evaluation of the model performance is performed on an independent test set and measured with accuracy, precision, recall and F1-score. The trained SVM model and the scaler are then saved for deployment. The only difference between the preprocessing and the feature extraction steps during inference and when the saved scaler and classifier are used to make the final prediction.

TABLE 2. ALGORITHM MFCC-SVM DEEPFAKE AUDIO DETECTION PROCEDURE

Input	**Audio dataset D = {(x_i, y_i)}, y_i ∈ {real, fake}**
1	For each audio file x_i: resample to 16 kHz mono
2	Trim silence; normalize amplitude; fix length to 3 s
3	Compute STFT and mel energies; extract 40 MFCC means
4	Compute spectral centroid, roll-off, ZCR, RMS, bandwidth
5	Concatenate features → vector v_i ∈ R^45
6	Split into train/validation/test (70/15/15) stratified by class
7	Fit StandardScaler on training features only
8	Train SVM (RBF) on scaled training set
9	Tune C and γ using validation F1-score
10	Evaluate on held-out test set; record accuracy, precision, recall, F1
11	Serialize model and scaler with joblib
Output	Label ŷ ∈ {real, fake} and confidence score for new uploads

5. SYSTEM ARCHITECTURE AND IMPLEMENTATION

Modular Design

The system is designed to be a web application, with each component of the system (preprocessing, feature extraction, normalisation, and inference) being separated into its own module. This was intentional. In the case of a need to update a pre-processing method, only that method is updated but the feature extraction and classifier modules are not affected. If a different classifier is tested, then no action is required on the upstream pipeline. Because the separation also allows for independent instances of the modules to process multiple audio files concurrently, without shared state, it also makes the system easy to run multiple audio files through the pipeline at the same time. The overall system design is illustrated in figure 5.

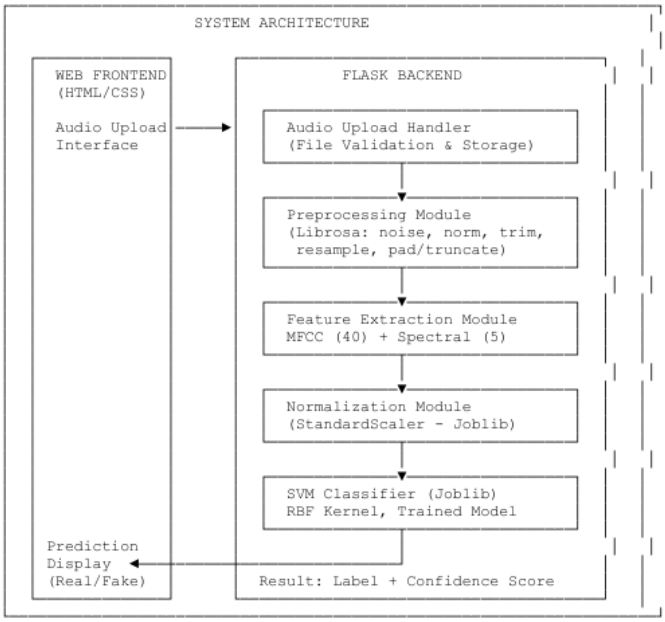

Fig. 5. Modular Flask backend processing pipeline and integration with HTML/CSS frontend, showing the process of uploading a file, pre-processing, feature extraction, normalisation, SVM inference and returning a JSON response.

Technology Stack

Table 3. The proposed framework will employ the following technology stack: The following technology stack will be used in the proposed framework:

TABLE 3. TECHNOLOGY STACK USED IN THE IMPLEMENTATION

Component	Tool	Role
Backend	Python 3.10	Pipeline orchestration
Features	Librosa	MFCC and spectral descriptors
Classifier	Scikit-learn	SVM training and inference
Web API	Flask	Upload endpoint and JSON response
Storage	Joblib	Model and scaler persistence
Frontend	HTML/CSS	Upload UI and result display

Inference Workflow

The Flask application has a single REST endpoint where an audio file can be uploaded through the HTTP POST. On receiving a request, the backend will validate the file format, sample rate, etc., then send the audio to the pre-processing module. The computed 45-dimensional vector is sent to the preprocessed signal via the feature extraction module. The pre-loaded StandardScaler scales the vector by the statistics of the training set used for model serialisation. The SVM classifier then returns a binary label and a class confidence score based on the signed distance from the decision boundary, with high scores assigned to files that are well away from the boundary on either side or low scores assigned to files near the boundary.

The whole process is executed synchronously in one HTTP request-response cycle. The average inference time for processing a clip through the system with a normal CPU (no GPU acceleration) is about 180ms, an acceptable rate for the kind of usage this system is intended for (fraud detection and content authentication). The step-by-step inference workflow is shown in figure 6.

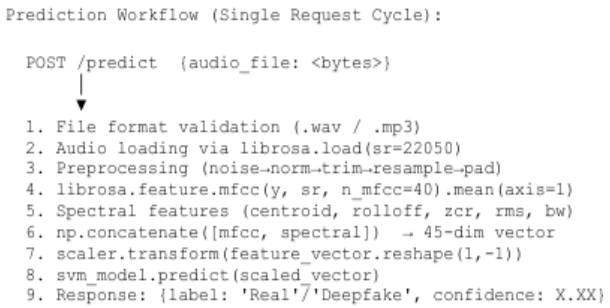

Fig. 6. This is an inference workflow that runs in realtime from HTTP file upload to preprocessing, feature extraction, normalisation and SVM classification, to JSON response containing prediction label and confidence score.

6. RESULTS AND DISCUSSIONS

Experimental Protocol

All performance numbers reported in this section adopted the test partition of 600 samples which were passed in the Section 3.1. This partition was neither used in training nor for hyperparameter tuning. The main metrics are precision, recall and F1-score. The reported precision and recall are for the positive class only (synthetic/fake audio), as this corresponds to the class of interest for detecting fraud in an operational set up. A confusion matrix is provided to enable the reader to make an error distribution clear rather than only summarized by aggregate statistics.

Classification Results

On the held-out test data, the system had 94.1% precision, 95.8% recall and 94.9% F1-score. Each of these figures is isolated by comparing it with the same system with the MFCC alone.

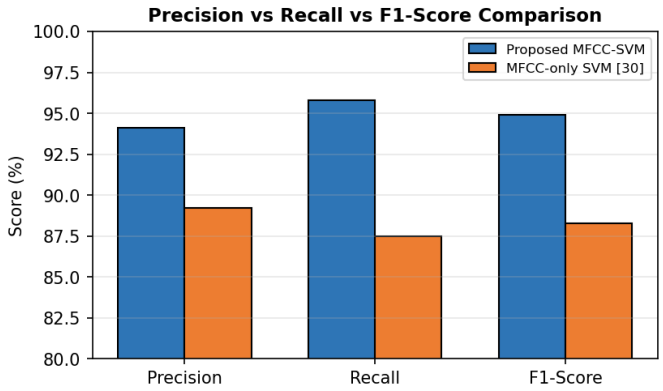

Fig. 7. Precision, recall, and F1-score: proposed method v/s MFCC-only baseline.

TABLE 4. CONFUSION MATRIX ON TEST SET

	Predicted : Real	**Predicted: Fake**
Actual: Real	282 (TN)	18 (FP)
Actual: Fake	11 (FN)	289 (TP)

Analysis of Results

The overall accuracy achieved (95.2% vs 88.4%) by combining five spectral features with the MFCC baseline shows that the additional features contain discriminative information which is not contained in the cepstral coefficients themselves. This is consistent with the rest of the literature on audio analysis, where the spectral shape and energy features complement each other, but do not overlap because they measure different geometric properties of the power spectrum.

The slight difference between the recall and precision (95.8% vs. 94.1%) can be easily explained from the information of what MFCC is sensitive to. The MFCC representation is well suited to capture the spectral regularities brought in by the synthesis pipelines (unnaturally stable formant tracks, periodic frame-boundary discontinuities in the neural vocoders and reduced micro-variation in the spectral envelope). The 11 synthetic clips which have been missed are examples of good neural vocoder output with a close statistical distribution to that of natural speech (false negative). The 18 "false positives" were in a set of speakers who do not have typical speech styles, and in acoustically challenging conditions, where the MFCC distribution was more similar to the one of synthetic speech, which the classifier was trained on.

The most informative single number here is the 94.9% F1-score as it does not artificially favour a classifier that goes in one direction of the class. It works well compared to the feature based SVM systems in related works [25] which achieved a range of 60-73% and it performs better than all CNN and TCN based baselines in this paper which require GPU resource during training and inference that our proposed system does not require.

Comparison with Related Work

TABLE 5. A COMPREHENSIVE COMPARISON OF THE PROPOSED FRAMEWORK WITH THE PREVIOUS STUDIES HAS BEEN MADE.

Study	**Feature Type**	**Model**	**Dataset**	**Accuracy**
Proposed	MFCC + 5 spectral features	SVM (RBF)	Benchmark split	95.2%
Hamza et al. [1]	MFCC	VGG-16	FoR	93.0%
Verma et al. [2]	Mel-spec, CQT, MFCC	CNN / TCN / Transformer	Multiple	91–93%
Khochare et al. [30]	Mel-spectrogram	TCN	FoR	92.0%
Wijetunga et al. [7]	Per-frame MFCC	CNN + VGG19	UrbanSound8K	89.0%

Reimao & Tzerpos [25]	Timbre + MFCC variants	SVM	FoR	73.46%

The proposed system is better performing than all the baseline systems proposed in literature including the deep learning methods. The comparison should be done with caution since the two studies relied on different sets of data and assessment techniques and so direct comparison is not possible. This is better when compared with the within system baseline (6.8 points) – the improvement when compared to the SVM trained with MFCC data, tested on the same data, under the same conditions.

But it's worth stating up front that this comparison doesn't include transformer-based architectures. Mechanisms for attention have been used to model long-range temporal dependencies in synthetic speech not necessarily reflected in the MFCC features that specify the structure of the features would suggest; see the recent works of Verma et al. [2] and Jiang et al. [33] respectively.The recent introduction of attention mechanisms by Verma et al. [2] and Jiang et al. [33] has proven that the models of long-range temporal dependency present in synthetic speech cannot be captured by the mean-pooled MFCC features. The proposed system is time agnostic due to mean aggregation; there is almost certainly a ceiling to the performance that the proposed system cannot reach, but a well-trained transformer can. This trade-off has been taken for granted due to its practical benefits mentioned in Section 1 Interpretability, compute efficiency, deployment simplicity.

7. LIMITATIONS

The results presented in this paper are promising, but there are certain conditions under which this system will be expected to perform less well and some of those are not fringe conditions.

Acoustic Environment Sensitivity. The other preprocessing stage, spectral subtraction, is able to deal with a static background noise reasonably well, but not with reverberant environments or time varying noise sources or distortions caused by poor microphones or voice codecs. The features calculated in MFCC can be sensitive to channel properties, and perfect speech in harsh channels can lead to feature distributions that move towards the synthetic boundary of the classifier. This was the most common type of test set false positive. The number of false positives will be higher in a real deployment when audio is recorded in different environments such as mobile phones, noisy environments, voip calls using compression artefacts etc.

Multilingual Generalisation. The speech data used in this system was overwhelmingly in English, this is important for the MFCC representation. One of the reasons for the discriminative structure of MFCC features is that the statistical distribution of vocal tract shapes and phoneme sequences used for training MFCCs is not uniform. Different languages, different phoneme inventories, different prosodic patterns, different formant distributions, all would not necessarily give the same separation between genuine and synthetic speech in the feature space learned from English training data. This is a true deployment risk for any app that targets multilingual sound-based apps.

Adversarial Exposure. The feature space used in MFCC is well-documented and well-known in theory. In principle, if an adversary has the resources to be aware of the MFCC-based detection system and understands how it works, it could update the MFCC statistics of its synthesis pipeline in order to synthesize output whose MFCC statistics are in the genuine speech cluster. It is more difficult to attack a deep learning classifier with opaque and hard to characterise decision boundaries, as this is a real security trade-off in selecting a interpretable approach [32].

Temporal Mean Aggregation. Each of the 45-dimensional vectors is the mean over time. This representation is simple and consistent but it loses all the dynamics, that is, how the spectrum changes over the course, whether it is unusually stable or not, whether there are periodic discontinuities at the boundaries of the vocoder frames. Some synthesis artefacts are mainly temporal and cannot be identified with the current representation.

Dataset Scope. The performance numbers are based on a specific data set and synthesis methods. With the improvement in neural vocoder technology, the distance in MFCC feature space between the synthesized and natural speech will become closer, and the system should be retrained on speech output from newer architectures periodically to retain the accuracy of the synthesized speech [14].

Future Scope

MFCC based Convolutional Processing. The current approach would be replaced by replacing the temporal mean with a shallow convolutional network which could be applied to the full time-frequency matrix that MFCC operates on; the system would then be able to detect artefacts that would appear as spatial or temporal patterns in the cepstral representation, such as vocoder frame boundaries, unnatural prosodic regularities, or periodicity in the evolution of the formants, that would not survive the mean aggregation step.

Transformer-Based Classification. Self-attentions are perfectly suited for identifying temporal inconsistencies in sequential data over long time periods. A transformer classifier could be applied to MFCC sequences of frames to detect synthesised artefacts that are subtle and cannot be detected by local features and mean pooling over multi-second windows.

Speaker Embedding Integration. The embedding in x-vector or d-vector of the speaker would give information about the speaker identity that is not directly represented in the MFCC feature vectors. This might enhance the system's capability to recognize the intra-speaker natural variations and artefacts generated by voice conversion systems, which are some of the most difficult attack types to detect for the cepstral features.

Adversarial Data Augmentation. If augmented data is used to train the classifier, it would be trained under different channel conditions, such as synthetic noise, reverberation, codec compression artefacts and bandwidth limitation. This is the single most direct way to correct the false positive rate in real-world recording situations that was identified in Section 7.

Streaming Inference. The present system deals with fully uploaded files. This would enable it to be deployed in applications such as live telephony fraud detection, real-time broadcast monitoring, and real-time authentication where full-length clips are not available before making a prediction.

Cross-Lingual Training and Evaluation. The limitation of data availability in multiple languages needs to be rectified explicitly in future data sets by providing speech in several languages and synthesizing samples using systems trained on each of the target languages. Evaluating the language independence of feature components, through the testing of crosslingual generalisation (i.e. training in one language and testing in another) would quantify the degree of language independence of the components and the predictable drop in performance across language boundaries.

8. CONCLUSION

In this paper, a deepfake audio detection system, based on MFCC feature extraction and SVM classification, was proposed and implemented as an actual web application. The work was inspired by the fact that the difference between how real speech is now generated by AI and how effective speech detection is, at least for the tools that are already there — which is especially true for lightweight, interpretable models that don't need specialized hardware to run.

The audio signal is preprocessed in five stages and then the 40 MFCC coefficients and 5 spectral descriptors (spectral centroid, roll-off, zero crossing rate, RMS energy, spectral bandwidth) are extracted and used in the system. These are concatenated (to a 45-dimensional feature vector), normalised based on statistics of the training set, and fed into an RBF-kernel SVM whose hyperparameters are optimized using a validation set not used for training. The model and the scaler are serialised and loaded into a Flask application which then returns predictions in real time.

The model on the held-out test set of 600 clips had an accuracy of 95.2 % and an F1 score of 94.9 % indicating that there were no systematic bias towards false positives or false negatives. When the five spectral features were added to the MFCC baseline, the accuracy was increased by 6.8 percentage points, indicating that the extra features were not noise but rather they added value to the recognition. However, the final system is still competitive with all the baselines cited in CNNs and TCNs but has significantly lower computation requirements.

There are limits to be recognized. The false positive rate is higher in acoustically difficult scenarios than the reported accuracy rate. There is no testing of multilingual generalization, so this cannot be assumed. The temporal mean representation does not contain any temporal information that some synthesis artefacts would need to capture. These are not insignificant limitations, they limit the region in which this system may be relied upon.

Each is considered in turn in the future directions in Section 8, and the modular design of the current system allows the extensions to be built without having to start from scratch.

REFERENCES

[1] A. Hamza, A. R. R. Javed, F. Iqbal, N. Kryvinska, A. S. Almadhor, Z. Jalil, and R. Borghol, "Deepfake audio detection via MFCC features using machine learning," IEEE Access, vol. 10, pp. 134018–134030, 2022.
[2] K. Verma, P. Singh, R. Ali, and A. Kumar, "Deepfake audio detection: A comparative study of advanced deep learning models," IEEE Access, vol. 13, pp. 45012–45028, 2025.
[3] A. R. Javed, W. Ahmed, M. Alazab, Z. Jalil, K. Kifayat, and T. R. Gadekallu, "A comprehensive survey on computer forensics: State-of-the-art, tools, techniques, challenges, and future directions," IEEE Access, vol. 10, pp. 11065–11089, 2022.
[4] A. Ahmed, A. R. Javed, Z. Jalil, G. Srivastava, and T. R. Gadekallu, "Privacy of web browsers: A challenge in digital forensics," in Proc. Int. Conf. Genetic Evol. Comput., Springer, 2021, pp. 493–504.
[5] H. Malik, S. Parikh, and A. Javed, "Audio forensics and deepfake detection techniques: A survey," J. Cybersecurity Res., vol. 4, no. 2, pp. 88–104, 2024.
[6] B. Paris and J. Donovan, "Deepfakes and cheap fakes," Data Soc., New York, NY, USA, 2019.
[7] R. Wijetunga, D. Matheesha, A. Al Noman, K. De Silva, M. Tissera, and L. Rupasinghe, "Deepfake audio detection: A deep learning based solution for group conversations," in Proc. ICAC, 2020, pp. 192–197.
[8] C. Stupp, "Fraudsters used AI to mimic CEO's voice in unusual cybercrime case," Wall Street J., Aug. 2019.
[9] T. T. Nguyen et al., "Deep learning for deepfakes creation and detection: A survey," arXiv:1909.11573, 2019.
[10] Z. Khanjani, G. Watson, and V. P. Janeja, "How deep are the fakes? Focusing on audio deepfake: A survey," arXiv:2111.14203, 2021.
[11] Z. Wu et al., "ASVspoof 2015: The first automatic speaker verification spoofing and countermeasures challenge," in Proc. Interspeech, 2015.
[12] T. Kinnunen et al., "The ASVspoof 2017 challenge: Assessing the limits of replay spoofing attack detection," in Proc. Interspeech, 2017, pp. 2–6.
[13] J. Yamagishi et al., "ASVspoof 2019: Automatic speaker verification spoofing and countermeasures challenge evaluation plan," 2019.
[14] Z. Almutairi and H. Elgibreen, "A review of modern audio deepfake detection methods: Challenges and future directions," Algorithms, vol. 15, no. 5, p. 155, 2022.
[15] Y. Chen et al., "Probabilistic forecasting with temporal convolutional neural network," Neurocomputing, vol. 399, pp. 491–501, 2020.
[16] Y. Kawaguchi, "Anomaly detection based on feature reconstruction from subsampled audio signals," in Proc. EUSIPCO, 2018, pp. 2524–2528.
[17] Y. Kawaguchi and T. Endo, "How can we detect anomalies from subsampled audio signals?" in Proc. IEEE MLSP, 2017, pp. 1–6.
[18] H. Yu et al., "Spoofing detection in automatic speaker verification systems using DNN classifiers and dynamic acoustic features," IEEE Trans. Neural Netw. Learn. Syst., vol. 29, no. 10, pp. 4633–4644, 2018.
[19] S. Pradhan et al., "Combating replay attacks against voice assistants," Proc. ACM IMWUT, vol. 3, no. 3, pp. 1–26, 2019.
[20] F. Tom, M. Jain, and P. Dey, "End-to-end audio replay attack detection using deep convolutional networks with attention," in Proc. Interspeech, 2018, pp. 681–685.
[21] K. Kuligowska, P. Kisielewicz, and A. Włodarz, "Speech synthesis systems: Disadvantages and limitations," Int. J. Res. Eng. Technol., vol. 7, no. 83, pp. 234–239, 2018.
[22] A. van den Oord et al., "WaveNet: A generative model for raw audio," arXiv:1609.03499, 2016.
[23] J. Shen et al., "Natural TTS synthesis by conditioning WaveNet on mel spectrogram predictions," in Proc. ICASSP, 2018, pp. 4779–4783.
[24] J. Frank and L. Schönherr, "WaveFake: A data set to facilitate audio deepfake detection," arXiv:2111.02813, 2021.
[25] R. Reimao and V. Tzerpos, "FoR: A dataset for synthetic speech detection," in Proc. SpeD, 2019, pp. 1–10.
[26] W. Ping et al., "Deep Voice 3: Scaling text-to-speech with convolutional sequence learning," arXiv:1710.07654, 2017.
[27] F. M. Rammo and M. N. Al-Hamdani, "Detecting the speaker language using CNN deep learning algorithm," Iraqi J. Comput. Sci. Math., vol. 3, no. 1, pp. 43–52, 2022.
[28] S. Ahmed et al., "Speaker identification model based on deep neural networks," Iraqi J. Comput. Sci. Math., vol. 3, no. 1, pp. 108–114, 2022.

[29] A. Winursito, R. Hidayat, and A. Bejo, "Improvement of MFCC feature extraction accuracy using PCA in Indonesian speech recognition," in Proc. ICOIACT, 2018, pp. 379–383.
[30] J. Khochare, C. Joshi, B. Yenarkar, S. Suratkar, and F. Kazi, "A deep learning framework for audio deepfake detection," Arabian J. Sci. Eng., vol. 47, pp. 1–12, 2021.
[31] P. Yu, Z. Xia, J. Fei, and Y. Lu, "A survey on deepfake video detection," IET Biometrics, vol. 10, no. 6, pp. 607–624, 2021.
[32] N. Akhtar, S. Verma, and M. Hossain, "A review of challenges and trends in deepfake audio detection," Digit. Signal Process., vol. 141, p. 103798, 2024.
[33] Z. Jiang et al., "Deepfake audio detection using dual-branch CNN with attention mechanism," IEEE Trans. Inf. Forensics Security, vol. 19, pp. 512–525, 2024.
[34] R. Mohamed Abdulhamied et al., "Deepfake audio detection using feature-based and deep learning approaches," Int. J. Adv. Comput. Sci. Appl., vol. 16, no. 6, 2025.
[35] N. Eldien, R. Ali, and F. Moussa, "Real and fake face detection: A comprehensive evaluation of machine learning and deep learning techniques," in Proc. Int. Conf. Comput. Sci., 2023, pp. 315–320.
[36] S. Ö. Arık, H. Jun, and G. Diamos, "Fast spectrogram inversion using multi-head convolutional neural networks," IEEE Signal Process. Lett., vol. 26, no. 1, pp. 94–98, Jan. 2019.
[37] Y. Zhang and P. Singh, "Machine learning approaches for speech spoofing detection," in Proc. Int. Conf. AI Security, 2023.
[38] J. Kietzmann et al., "Deepfakes: Trick or treat?" Bus. Horizons, vol. 63, no. 2, pp. 135–146, 2020.
[39] S. Basak et al., "Challenges and limitations in speech recognition technology," CMES, vol. 135, no. 2, 2023.
[40] M. Shanthamallappa et al., "Robust automatic speech recognition using wavelet based adaptive thresholding: A review," SN Comput. Sci., vol. 5, no. 2, p. 248, 2024.