DOI: 10.2478/ijasitels-2025-0003

Deepfake Voice Detection for Underrepresented Languages: A Romanian Case Study

Bălășoiu Robert-Alin[1], Ivașcu Ioana-Daniela[2] , Mihu Cantemir[2], Muntean Robert-Andrei[2], Olescu Marco Leon[3], Resiga Sorana-Ioana[4]

[1]*Politehnica University of Timisoara, Romania*

[2]*Lucian Blaga University of Sibiu, Romania*

[3]*NTT DATA Romania*

[4]*Babes Bolyai University of Cluj Napoca, Romania*

Abstract

The rapid advancement of artificial intelligence and its subsequent application in deepfake media present a significant global security concern. Despite this widespread issue, effective detection solutions are notably absent for less commonly used languages. This paper proposes a potential solution for identifying generated audio in Romanian. The solution centres on an SVM-based algorithm, previously demonstrated to perform effectively in English language tests, adapted with a dataset specifically tailored to the Romanian language. The resulting language-specific model exhibits better performance in differentiating between authentic and synthetic Romanian audio, thereby offering an improvement over general-purpose, Anglocentric systems. This constitutes an effective strategy for developing localised solutions for a language with limited resources.

Keywords: Deepfake detection, Synthetic speech, Romanian language, Support Vector Machine (SVM)

1. Introduction

Artificial intelligence has been one of the most researched topics in recent history. This field has experienced an exponential growth, faster than any of its predecessors; thus, it has created a new cluster of problems that we are still trying to grasp. One of the most pressing issues is the rise of deepfake media. While it existed long before through digital manipulation, AI makes it much more accessible and realistic than ever before. The dangers of it have been thoroughly studied and analysed, but more importantly, they have been felt as a shockwave in our society. Because of its implications in both social and personal spaces, it became imperative that we find tools to mitigate this threat.

Current research in the domain has already identified various methods for handling this problem, from watermarks embedded in the content as it is created to tools that assess the media and assign it a likelihood of being generated by AI. One problem encountered

is the lack of accuracy of these tools for underrepresented languages [1]. English, Spanish, German, and other widely used languages have the advantage of extensive training databases, allowing these tools to be honed and perfected. Unfortunately, Romanian is not among these languages.
The purpose of this paper is to explore findings on this language: how pre-existing tools perform in detecting audio generated by AI, as well as a proposed model tailored to handle this language. The different models that have been tested underperformed, producing false positives on audio that a native speaker would easily categorise correctly. Because of that, a Support Vector Machine (SVM) model has been trained on two databases, MLAAD[2] for generated audios, and UPSDATRO [3] for real audio. Such a focused approach has managed to classify correctly between real and fake audio, especially in a controlled test environment. Despite that, the shortcomings caused by a lack of training data leave room for improvement when it comes to scaling the model and increasing accuracy.
This study confirms that developing targeted, language-specific models is an important advancement in audio detection for languages with limited data resources. Yet, the main problem is caused by the lack of data that can be used for training and testing such models.

2. The AI Detection Model

The proposed detection algorithm was implemented in Python, using librosa, scikit-learn, and joblib for audio processing, classification, and model persistence. The system architecture is designed around a Support Vector Machine (SVM) classifier, trained to distinguish between AI-generated and authentic Romanian speech using Mel-Frequency Cepstral Coefficients (MFCCs) [4] as acoustic features. The implementation is adapted from an open-source project [5].

2.1. Data Processing and Feature Extraction

Before classification, the model converts each audio file into a compact feature representation using MFCCs. MFCCs are among the most widely used feature sets in speech analysis, as they approximate the human ear's sensitivity to frequency and capture a voice's timbral characteristics.
The function `extract_mfcc_features()` loads each .wav file using librosa, computes 13 MFCC coefficients per frame (`n_mfcc=13`), and returns the mean of all frames to form a 13-dimensional vector per sample. This ensures a consistent input size regardless of the audio file's duration.

Listing 1: MFCC Feature Extraction

```
mfccs = librosa.feature.mfcc(y=audio_data, sr=sr,
                             n_mfcc=n_mfcc,
                             n_fft=n_fft,
                             hop_length=hop_length)

return np.mean(mfccs.T, axis=0)
```

This process is identical during both training and inference, ensuring that the model interprets new data consistently.

2.2. Dataset Integration

Two datasets were used to train and evaluate the model:

- **UPSDATRO** – a curated collection of *authentic Romanian speech* samples, serving as the positive (real) class.
- **MLAAD (Machine Learning Audio Anti-Deepfake)** – a collection of *AI-generated* Romanian voices created by text-to-speech (TTS) systems such as Bark or multilingual models, representing the synthetic (fake) class.

Each dataset directory is recursively scanned for .wav files, which are processed to extract MFCCs and labelled accordingly (0 for real, 1 for fake). The combined dataset is then split into training and testing subsets (80/20 ratio) using `train_test.split()` with stratification to preserve class balance.

2.3. Model Architecture

The classifier is a **Support Vector Machine (SVM)** [6] implemented through sklearn.svm.SVC. A linear kernel was selected for its balance between performance and interpretability, and probability estimates were enabled to provide confidence scores during inference.

Listing 2: SVM Classifier Initialisation

```
svm_classifier = SVC(kernel='linear', random_state=42, probability=True)
svm_classifier.fit(X_train_scaled, y_train)
```

Before training, all features are standardised using StandardScaler() to ensure uniform scaling across dimensions, which is crucial for SVMs to compute meaningful margins between classes

2.4. Model Training and Evaluation

Training involves fitting the SVM to the normalised training data and evaluating it on the test set. The model's performance is measured using:

- **Accuracy**: proportion of correctly classified samples.
- **Confusion matrix**: a detailed view of true vs. false predictions for both real and fake classes.

Listing 3: Model Evaluation Metrics

```
accuracy = accuracy_score (y_test,
y_pred )
confusion_mtx = confusion_matrix
(y_test, y_pred )
```

The trained model and scaler are serialised using joblib as svm_audio_classifier.pkl and scaler.pkl, respectively, allowing efficient reuse without retraining.

2.5. Inference Pipeline

For inference, the trained model loads once at runtime:

Listing 4: Model Loading for Inference

```
svmmodel = joblib.load(" svm_audio
classifier .pkl") scaler =
joblib.load("scaler.pkl")
```

Given a new Romanian audio file, the system:

1. Extracts MFCC features.
2. Normalises them using the saved scaler.
3. Uses the SVM's predict proba() method to estimate class probabilities.

The output includes a predicted label ("Real" or "Fake") and a confidence score expressed as a percentage:

Listing 5: Inference Output Generation

```
prediction_proba = svm_model. predict_proba (
features_scaled )[0] confidence = prediction
proba [ prediction_index ] *100
```

2.6. Implementation Summary

Table 1. Summary of Model Implementation

Stage	Description
Feature Extraction Classifier	MFCCs (13 coefficients) computed with librosa
Preprocessing	Support Vector Machine (SVC, linear kernel, probability=True)

Datasets	Standardisation using StandardScaler UPSDATRO (real), MLAAD (fake)
Evaluation metrics Output	Accuracy, Confusion Matrix “Real” or “Fake” + confidence percentage
Model persistence	Saved using joblib

This lightweight, interpretable approach enables high performance even with limited linguistic data, providing a scalable framework for language-specific audio authenticity detection.

3. Experimental Works and Results

3.1. Databases and Training Configuration

To evaluate whether a classical machine-learning method can effectively detect Romanian deepfake speech, we used two publicly available audio datasets representing both authentic and synthetic Romanian voices. For genuine speech, we relied on the **UPSDATRO dataset**, a curated corpus containing **1,001 recordings** of native Romanian speakers. These recordings include a variety of voices, dialects, genders, and background conditions, which provide natural acoustic diversity. For synthetic data, we used the **MLAAD dataset**, which contains **2,637 AI-generated Romanian audio samples** created using multiple modern text-to-speech (TTS) systems. This dataset is part of a multilingual anti-spoofing collection specifically designed for deepfake research.

The choice of these two datasets was guided by the fact that **Romanian is a low-resource language**, with few open datasets for voice anti-spoofing research. Many of the most common deepfake detection models are trained primarily on English or Mandarin, where massive speech corpora are available. However, transferring such models directly to Romanian often results in poor performance due to phonetic and prosodic differences. Thus, our goal was to test whether even a relatively small, language-specific dataset could produce a reliable detection model.

Although the two datasets differ in size, combining them created a **diverse yet balanced training foundation**. Altogether, the combined dataset contained **3,637 samples (1,001 real and 2,637 fake)**. Before training, all audio files were standardised in format, converted to mono, and resampled to a consistent rate to minimise potential artefacts caused by inconsistent sampling frequencies. The data were randomly shuffled and split into **80% for training (2,909 samples)** and **20% for testing (728**

samples), following a common machine-learning convention that reduces overfitting and ensures fair generalisation testing.

To further investigate how dataset structure affects learning, a **second experiment** was conducted with a perfectly balanced corpus by randomly selecting **1,000 real and 1,000 fake recordings**. This setup allowed us to examine whether class balance impacts model generalisation. The same 80/20 ratio was applied, resulting in **1,600 training samples (800 real, 800 fake)** and **400 testing samples (200 real, 200 fake)**. This comparative design helped identify whether the imbalance of the initial configuration caused the model to bias toward either class.

Table 2. Training Configurations

Case	Real Samples	Fake Samples	Total
Initial configuration	1,001	2,637	3,637
Balanced configuration	1,000	1,000	2,000

In both scenarios, the model was trained using **short-form Romanian speech clips**, most of which were clean and of moderate duration. While this structure offered clear advantages for training consistency and computational simplicity, it also limited the exposure of the model to background noise, long speech sequences, and spontaneous conversational dynamics. These factors, as will be discussed later, played an important role in the real-world evaluation phase.

3.2 Audio Duration

The average duration of the audio clips used in training ranged between **2 and 10 seconds**, with most samples centred around **approximately 8 seconds**. This time range was not arbitrary; short-duration speech segments are common in audio classification studies because they capture enough acoustic variability for meaningful feature extraction while maintaining manageable computational complexity. Each clip generally contained a single spoken sentence or phrase, allowing the model to focus on the **core spectral and prosodic characteristics** of the voice without distractions from prolonged pauses or background interference.

Short recordings have both benefits and drawbacks. On the positive side, shorter clips make it easier for traditional machine-learning models, such as **Support Vector Machines (SVM)**, to process input data since these models rely on fixed-length feature vectors and do not model temporal dependencies directly. This makes MFCC-based feature extraction particularly effective, as it compresses the essential characteristics of speech into a compact representation. However, the drawback is that short clips may not include sufficient speech diversity. Variations in tone, rhythm, and articulation that

occur across longer recordings might not be captured, which can reduce model robustness when faced with unfamiliar speaking patterns.

In our dataset, **variability in duration influenced model confidence**. The SVM tended to perform best on clips with durations close to the average training length, as the MFCC features extracted from such samples were consistent with what the model had learned. Extremely short or very long recordings sometimes resulted in unstable predictions because the averaged MFCC features were less representative of typical speech behaviour. This observation suggests that future datasets should aim for greater uniformity in clip duration or employ segment-based approaches that split long recordings into smaller, consistent frames.

Furthermore, short-duration training data mirrors how Romanian speech often appears in real-world scenarios, in brief utterances from phone recordings, short social media posts, or voice assistants. By focusing on these smaller clips, our model simulated practical conditions where users might submit or encounter only a few seconds of speech for verification. However, as demonstrated in Section 3.4, when exposed to long, noisy, or mixed-content recordings, performance decreased significantly. This reveals that **duration variability is not just a technical detail but a major factor shaping generalisation and robustness**.

Table 3. Audio Duration Statistics

Metric	Value
Average Duration	8.32 s
Minimum Duration	0.44 s
Maximum Duration	43.58 s
Standard Deviation	6.78 s
Estimated Total Duration	13.0 h

3.3 Model Training and Evaluation

Our deepfake voice detector was implemented as a **lightweight pipeline** combining Python, **Librosa** for feature extraction, and **scikit-learn** for classical machine learning. Each audio sample was first processed to extract **MFCCs**, a compact representation of how sound energy is distributed across perceptually meaningful frequency bands. The MFCCs were then summarised by computing the **mean of 13 coefficients**, producing a single fixed-length feature vector per recording. This step simplified the input structure while preserving enough acoustic information for distinguishing real and synthetic speech.

Before training, we applied a **StandardScaler** to normalise all features, ensuring that each dimension contributed equally to the learning process. The model was then trained using an **SVM** with a linear kernel. The linear SVM was chosen for three main reasons:

1. It performs well with small and medium-sized datasets, making it ideal for low-resource languages.

2. It offers interpretability by providing clear decision boundaries between real and fake samples.
3. It is computationally efficient, requiring no GPUs and minimal parameter tuning.

For both dataset configurations, an **80/20 split** was used for training and testing. The initial configuration produced **2,909 training samples** and **728 testing samples**, while the balanced configuration included **1,600 training samples** and **400 testing samples**.

Table 4. Initial Split

Set	Train	Test
Real (initial)	800	201
Fake (initial)	2,109	528
Total	**2,909**	**728**

Table 5. Balanced Split

Set	Train	Test
Real (balanced)	800	200
Fake (balanced)	800	200
Total	**1,600**	**400**

The results obtained in controlled testing were **remarkably high**. The model trained on the larger, unbalanced dataset achieved **100% test accuracy**, while the balanced version achieved **99.5%**, with only a few misclassifications. The confusion matrices revealed near-perfect separation between real and fake speech, indicating that the SVM successfully captured distinguishing MFCC-based patterns.

However, such high accuracy must be interpreted cautiously. These values were achieved in a **lab-like environment**, using clean data closely resembling the training samples. While this confirms that the SVM learned to differentiate synthetic from genuine speech under controlled circumstances, it does not necessarily imply the same level of performance in uncontrolled settings. **While these results demonstrated strong classification capability in a controlled setup, the next step was to test whether such performance holds in an uncontrolled, real-world environment.**

3.4 Real-World Evaluation

To assess how the trained model behaves with natural, uncontrolled audio, we performed a **real-world evaluation** using **60 Romanian TikTok clips** and manually collected samples from various online sources. The dataset contained **30 short clips (2–**

5 seconds) and **30 long clips (approximately 8 seconds to over 3 minutes)**, covering a mix of real and AI-generated voices. This test simulated how a detection system might operate in practice, analysing user-uploaded or spontaneous speech data with variable background noise and compression.

When evaluated on this real-world dataset, the model's behaviour diverged significantly from its laboratory results. While our testing yielded an average accuracy of 45%, reflecting the challenge of analyzing raw, unfiltered data that contain real-life interferences, this still represents a significant improvement over our initial tests on the multipurpose models (Gemini, Claude, ChatGPT), which averaged only 20% in the same scenarios. On short clips, it correctly recognised all real recordings but **misclassified nearly all synthetic ones as real**, suggesting that speech generated by modern TTS systems closely mimicked the acoustic cues the SVM had learned to associate with authenticity. For longer clips, the issue became more pronounced — **the vast majority of samples, whether real or fake, were classified as real.**

This bias toward the "real" class suggests that the model's decision boundary, learned from clean MFCC averages, may **underfit the broader variability** of real-world audio. Background sounds, room echo, speech overlap, and microphone differences altered the spectral structure of the audio in ways that the model had never encountered during training. These effects weakened its ability to recognise subtle synthesis artefacts.

Despite the drop in accuracy, these tests were highly instructive. They highlighted that real-world recordings introduce **domain shift**, a condition where test data differ systematically from the training data. In the context of deepfake detection, domain shift can stem from environmental noise, variable speaking styles, or compression artefacts typical of social media platforms. Understanding this mismatch is essential for improving model robustness. A practical next step would involve collecting more diverse samples or augmenting existing data with simulated noise and distortions to make the system better prepared for real-world variability.

4. Experimental Observations and Conclusions

This research highlights both the strengths and limitations of classical machine-learning methods in Romanian deepfake voice detection. The experiments demonstrated that even a simple SVM classifier, when trained on well-prepared and balanced datasets, can achieve excellent performance under controlled conditions. This confirms that **traditional, interpretable models** can remain valuable tools for languages where deep-learning resources and large datasets are not yet available.

At the same time, the **performance gap between controlled and real-world evaluations** reveals an important limitation. The model struggled when exposed to unpredictable or noisy inputs, particularly long, spontaneous recordings. It tended to classify unfamiliar acoustic conditions as "real," suggesting that **MFCC-only features** are insufficient to capture the full range of synthetic voice artefacts in diverse

environments. The strong bias observed toward real samples underscores the need for richer, context-aware features or hybrid architectures.

Two key lessons emerged from these findings:

1. **Language-specific machine-learning models** can outperform generic, multilingual detectors when properly trained, even if they rely on classical techniques.
2. **Diverse and realistic datasets** are essential for robust real-world performance, particularly in underrepresented languages where data scarcity amplifies overfitting risks.

Future work will focus on expanding the dataset with additional speaker profiles, longer and noisier recordings, and a broader range of TTS-generated voices. Implementing **audio segmentation**, **data augmentation**, and **hybrid SVM-neural approaches** could further enhance adaptability. By combining the interpretability of SVMs with the feature-learning capabilities of neural networks, future systems could balance accuracy and efficiency while maintaining accessibility for Romanian researchers and institutions.

Ultimately, this project demonstrates that even in the absence of large-scale computational infrastructure, **meaningful progress can be achieved through careful dataset design, balanced experiments, and systematic evaluation**. The lessons learned from this Romanian case study may serve as a foundation for developing deepfake detection frameworks for other low-resource languages facing similar challenges.

References

[1] Joshi, Pratik, Sebastin Santy, Amar Budhiraja, Kalika Bali, Monojit Choudhury. *The State and Fate of Linguistic Diversity and Inclusion in the NLP World.*, Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 6282–6293, ISBN N/A, Online, 2020.
[2] Müller, Nicolas M., Piotr Kawa, Wei Herng Choong, Edresson Casanova, Eren Gölge, Thorsten Müller, Piotr Syga, Philip Sperl, Konstantin Böttinger. *MLAAD: The Multi-Language Audio Anti-Spoofing Dataset.*, arXiv preprint arXiv:2401.09512, ISBN N/A, Online, 2024.
[3] Păiș, Vasile, Verginica Barbu-Mititelu, Elena Irimia, Radu Ion, Dan Ioan Tufis. *USPDATRO: Under–Represented Speech Dataset from Open Data – Case Study on the Romanian Language.*, Zenodo Repository, DOI:10.5281/zenodo.7898232, ISBN N/A, Online, 2024.
[4] Ahmad, Wali Muhammad. *AI Voice-Detection (TensorFlow).*, GitHub Repository, https://github.com/WaliMuhammadAhmad/AIVoice-Detection/tree/master/tensorflow, ISBN N/A, Online, 2025.
[5] Sahidullah, Md., Tomi Kinnunen, Cemal Hanilçi. *A Comparison of Features for Synthetic Speech Detection.*, Proceedings of Interspeech 2015, 2087–2091, ISBN N/A, Dresden, Germany, 2015.
[6] Cortes, Corinna, Vladimir Vapnik. *Support-Vector Networks.*, Machine Learning, Volume 20, Issue 3, Pages 273–297, ISSN 0885-6125, Springer, New York, 1995.