iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 12 **Issue:** V **Month of publication:** May 2024

DOI: https://doi.org/10.22214/ijraset.2024.61718

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue V May 2024- Available at www.ijraset.com

Deepfake Audio Detection System

Tushar Kapoor[1], Abhi Gaur[2], Tushar Rajora[3], Yogya Prashar[4], Jyoti Parashar[5]
Dr. Akhilesh Das Gupta Institute of Professional Studies, Delhi, India

Abstract: ***To ensure the authenticity of audio material, it is important to establish reliable detection tools that can trace the spread and use of deepfake technologies. This study focuses on the methodology of deepfake audio identification via Mel Frequency Cepstral Coefficients (MFCC) as features with Random Forest as a classifier. By extracting MFCC features from audio clips and using them in a Random Forest model, one can learn unique spectral properties that can help distinguish deepfakes from authentic audio. The Random Forest algorithm, famous for its quality of being able to work well in an ensemble learning paradigm, is utilized to identify patterns that are representative of deepfake manipulation. To ensure the efficiency and reliability of this method, it was tested on a large number of different data sets that included both genuine and fake voice samples. To ensure robustness and generalization, cross-validation techniques are employed, restricting model predictions to the range of 0 to 1 and providing informative error messages for effective diagnosis. Thus, it is an important scientific study helping to develop and strengthen methods for identifying and eliminating threats in the area of artificial sound activity.***

Keywords: Audio Deepfakes, Audio detection, MFCC, Random Forest, Machine learning, Feature extraction, Fraud detection, Pattern recognition.

I. INTRODUCTION

The dominance of deepfake technologies in the recent past is introducing a new wave of media manipulation where authenticity and the truth of audiovisual content are faced with unmatched challenges. The application of deepfakes in the generation of super realistic synthetic media using high-level machine learning algorithms is a major menace that threatens sectors such as journalism, politics, entertainment, security, and others. Though a lot has been done on how to detect deepfake videos, little or nothing has been achieved in detecting manipulated audio, a challenge that carries far and wide consequences.

Future content would include voice cloning or speech synthesis that could end up tricking many people into believing falsehoods on a scale never before seen. An urgent problem, then, is the need to detect deepfake audio through the development of robust, reliable and applicable techniques. In short, deepfake audio requires specialised methods. Regular audio forensics methods can help, to an extent, though knowledge of the underlying data can expose certain signals. Yet these techniques are often limited with the more subtle signals associated with emerging deepfakes.

A. Deepfake Classification and Generation

1) Imitation-based Deepfakes

Imitation-based Deepfakes, a method of transforming speech to mimic another voice for purposes such as privacy protection, often involve advanced technology and mimicry techniques. Whether it's a voice impersonator using sophisticated tools or algorithms manipulating existing recordings, classifiers play a crucial role in unmasking such trickery. Spectral analysis, utilizing spectrograms to visualize sound frequencies, helps classifiers identify irregularities introduced during imitation. These anomalies include rapid shifts in formants, which shape vowels, and spectral artefacts—mathematical traces left in the frequency patterns due to manipulation. Moreover, imitation-based deepfakes can employ various techniques, from human imitation to masking algorithms like Efficient Wavelet Mask (EWM). These algorithms generate new audio that closely resembles the target voice, making it challenging for humans to discern between real and fake audio. By understanding these methods and utilizing sophisticated classifiers, we can better detect and combat the proliferation of imitation-based deepfakes.

2) Synthetic-based Deepfakes

Synthetic-based or Text-To-Speech (TTS) technology utilizes machine learning algorithms to generate new speech that closely resembles human voices. Detecting synthetic-based deepfakes involves identifying statistical irregularities from deep learning models and inconsistencies with the target's typical speaking style. TTS systems consist of modules for analyzing text, modelling acoustic features, and encoding speech. These models, trained on organized audio datasets, create authentic-sounding synthetic audio by replicating both the linguistic nuances of the input text and the characteristics of the target speaker. The intricate process highlights the difficulty in distinguishing synthetic-based deepfakes from genuine human speech.

3) Replay-based Deepfakes

Relay-based deepfakes involve merging bits of real audio into a new recording, which makes it hard to be detected. These deepfakes stitch together real snippets creating inconsistent features that can be easily seen by classifiers. This process of stitching results in acoustic discontinuities such as changes in background noise or reverberation from one segment to another. Besides, speech flow disruptions like sudden pace and emotion change display the points where segments were attached. In replay-based attacks, far-field and cut-and-paste methods are commonly used so that the target speaker's recorded audio is changed for malicious reasons.

B. Mel-Frequency Cepstral Coefficients (MFCC)

The conventional ways of storing reputations were difficult because of the challenges such as high dimensionality and variability in speech patterns, hindering the accuracy which compromised the reliability and generalization of reputation systems. Though the acclamation of this technology facilitated the development of the related and relevant disciplines, the concept of MFCC contributed to a groundbreaking revolution and gave accurate information representations resilient to a wide range of factors such as speakers' variations, historic noise, and speech pronunciation fluctuations.

One of the audio processing methods that is widely used is MFCC, which stands for Mel-Frequency Cepstral Coefficients. It involves changes in the way how frequency area representation of a speech signal can be encoded into a structured set of cepstral coefficients. This approach provides the tooling to effectively apply the characteristics of audio sound signals which makes it more possible to solve the challenging issues encountered in voice identification, speaker differentiation, and ideal audio detection tasks.

The computation of MFCC involves several steps:

Step 1: Windowing

- Audio signal is subdivided into short adjacent overlapping frames.
- For any frame, a window function with the window is being utilized.

Step 2: Mel Filtering

- For each frame power spectrum is being computed using Discrete Fourier Transform (DFT).
- The obtained spectrum, which is Mel-scaled, is passed through a filter bank whose controls conform to the scale.
- The energy of each filter bank is squared to yield the filter bank outputs.

Step 3: Logarithmic Scaling

- A logarithm is taken for each filter bank energy value to align with the logarithmic scale that serves for human auditory system.

Step 4: By Discrete Cosine Transform (DCT),

- DCT transforms log filter bank energies into a decorrelation function
- Discretely Cosine Transform (DCT) part coefficients which result in Mel-Frequency Cepstral Coefficients (MFCCs).

C. Random Forest

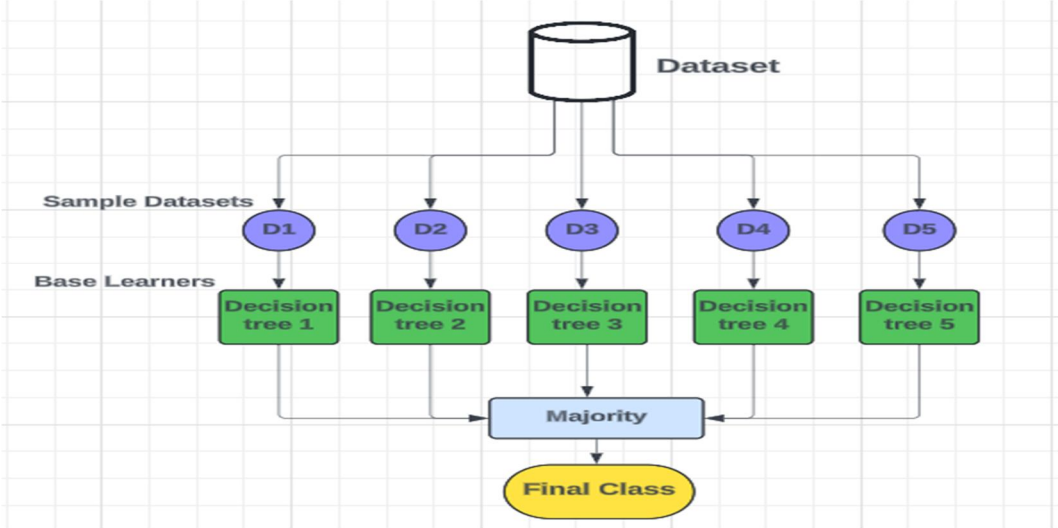
Random Forest or Random decision forest is a popular and adaptable ensemble learning method that is widely used for classification and regression tasks. It constructs an ensemble of decision trees during the training phase, where each tree is built independently. One of the main features of Random Forest is its dependence on randomness at various stages of the algorithm. Firstly, each decision tree is trained on a random subset of the training data that is sampled and feature selected with replacement. This process is known as bootstrap sampling, which helps create diversity among the trees and helps in alleviating overfitting. Moreover, at each different node of the decision tree, only a random subset of features is considered for splitting further enhancing the diversity among the trees and preventing them from depending too much on any single feature. In several classification tasks, the predictions are aggregated through the individual trees, however averaging in other regression tasks to achieve the final prediction of the random forest. This ensemble approach often results in higher accuracy and low error rate with high precision compared to individual decision trees.

Workflow of the Random Forest Algorithm

Step 1: Initially, a random subset of the training dataset is selected with replacement, a process known as bootstrap sampling.

Step 2: From this randomly selected dataset, a decision tree is constructed using a subset of features chosen randomly at each node.

Step 3: During prediction, each decision tree in the forest independently outputs a class prediction or a numerical value. The final prediction is determined by the class or value selected is the one chosen by the majority of trees.



II. LITERATURE REVIEW

[1] This review, authored by Aditi Govindu et al., underscores the significance of utilizing custom DCGANs for generating convincing fake audio samples and their potential impact on classification models. By introducing the FAD metric and incorporating XAI techniques like LIME, SHAP, and GradCAM, researchers gain valuable insights into classifier decision-making processes. These findings not only enhance the accuracy and efficiency of audio classification models but also highlight the need for ongoing research to develop more robust and secure deep learning systems, particularly in domains vulnerable to the malicious use of fake samples.

[2] In this study by Lanting Li et al., an advanced audio deepfake detection model was developed, featuring improvements in front-end feature extraction and back-end classification. Augmentation using a self-supervised model enhanced generic linguistic feature extraction, while α-FMS integration in RawNet2 improved feature map discrimination. Future research aims to integrate the system with speech recognition, explore diverse loss functions, and enhance effectiveness against various attacks. Additionally, efforts will focus on unlabeled detection techniques, such as self-supervised knowledge distillation, to yield pragmatic solutions for evolving forgery methodologies. Anticipating the ASVspoof 2023 challenge, ongoing vigilance ensures appreciation of advancements in voice deepfake detection.

III.PROPOSED FRAMEWORK

The framework for detecting for deepfake audio production uses the Python programming language and essential libraries, so forming a fully- functional detection system.

1) *Python Programming Language*: The Python language becomes the backbone on which the deepfake audio detection system relies on. Python, with its simplicity, flexibility, and comprehensive library backup, is suited for the tasks of machine learning and instantaneous designing and machine prototyping.
2) *Required Python Packages*:

- Numpy: Numpy represents a fundamental library for numerical computing as well as for array operations. It provides the best data structures to operate on large data sets and also gives the fastest performance in terms of mathematic operations.
- Soundfile: This package permits reading as well as writing audio files in a wide range of formats, providing the necessary support for audio data provisioning and intensity within the detection system.
- Librosa: Librosa, a library for audio signal processing, includes tools to load audio files, extract audio features (MFCC), generate waveform plots and spectrograms.
- Scikit-learn: An undoubted library for machine learning in Python is scikit-learn which offers a large variety of algorithms for classification, regression, clustering, etc. It provides features for data preprocessing, model evaluation, and cross-validation, which, in turn, makes the construction and testing of the model to detect deepfake audio quicker and easier.

IV. METHODOLOGY

The study uses the Random Forest classification technique to handle the problem of identifying deepfakes, or fake audio samples. The primary goal is to develop an intelligent system that can distinguish between genuine and fake audio files with efficiency. Moreover, Python contributes to over 78.5% of all code. Additionally used librosa and scikit-learn libraries to assist in the development of the deepfake detection system

A. *Data Collection and Preprocessing*

The data collection process involves gathering audio samples from two directories: one containing real audio files and the other containing deepfake audio files. These directories are specified as real_audio_dir and deepfake_audio_dir, respectively.
The audio samples from the designated directories are loaded using the load_data function. Using the extract_features function, it iteratively goes through each file in the directory, extracts features from the audio file, and adds the features along with the associated label to a list. The label is assigned in accordance with the function's label parameter.

The preprocessing steps include:

1) *Loading*: Loading audio files from the specified directories.
2) *Feature Extraction*: Extracting features from the audio files using the extract_features function (MFCC).
3) *Labeling*: Assigning labels to the extracted features based on the provided label parameter (0 for real audio, 1 for deepfake audio).

The extracted features and corresponding labels are stored in separate lists (features and labels). These lists serve as the basis for constructing the dataset used for training the deepfake audio detection system.

B. *Feature Extraction*

In order to prepare audio data for machine learning tasks, feature extraction is an essential step. In this case, the feature extraction approach is utilized to extract useful representations of audio signals from raw waveform data.

1) *Mel-Frequency Cepstral Coefficients (MFCC)*

Mel-Frequency Cepstral Coefficients (MFCC) are the main feature extraction method used. As MFCCs is highly effective at capturing the spectrum features of audio signals, they are frequently utilized in audio processing, especially for tasks like audio detection and speech recognition.

2) *Librosa Library*

MFCC features are computed using the librosa library, a popular Python package for audio and music signal processing. Librosa provides convenient functions for loading audio files, computing various audio features, and performing signal analysis.

The Extraction Process includes:

- The feature extraction process begins by loading the audio file using the librosa.load() function, which returns the audio waveform and its corresponding sampling rate.
- MFCC features are then computed using the librosa.feature.mfcc() function, which applies a series of signal processing techniques, including the Fast Fourier Transform (FFT), Mel filtering, and Discrete Cosine Transform (DCT), to extract a set of coefficients representing the spectral envelope of the audio signal.
- The resulting MFCC matrix typically consists of a series of feature vectors, with each vector representing the MFCC coefficients computed for a short segment of the audio signal.
- To ensure a fixed-length representation, the MFCC matrix is processed to truncate frames as necessary, ensuring consistency across different audio files.

A two-dimensional matrix representing the retrieved MFCC features is presented, with each row representing a frame of MFCC coefficients and each column representing a single MFCC coefficient.This matrix serves as the input data for training the Random Forest classifier, providing a compact and informative representation of the spectral characteristics of the audio signals.

C. Model Training

Model training is a pivotal stage in developing a deepfake audio detection system, where the Random Forest classifier is trained using the extracted MFCC features to distinguish between genuine and manipulated audio samples.

The Training process includes:

1) *Initialization*: The Random Forest classifier is initialized with hyperparameters such as the number of trees and random state for reproducibility.
2) *Training Process:* The training process involves fitting the Random Forest classifier to the input data, which consists of the MFCC feature matrix (X) and corresponding labels (y). The fit() method is called on the classifier object, passing X and y as arguments.
3) *Cross-Validation:* Cross-validation is performed to assess the generalization performance of the trained model and evaluate its robustness to variations in the training data. The cross_val_score() function from scikit-learn is used to compute the cross-validated accuracy of the model.
4) *Model Evaluation:* After training the Random Forest classifier, it is evaluated using performance metrics such as accuracy, precision, recall, F1-score, and area under the ROC curve (AUC). These metrics provide insights into the classifier's ability to correctly classify genuine and manipulated audio samples.
5) *Model Interpretability:* Random Forest classifiers offer interpretability through feature importance scores, which indicate the relative importance of each feature (MFCC coefficient) in making predictions. Feature importance scores can be visualized to gain insights into the discriminative power of different MFCC features.

V. IMPLEMENTATION

1) Import required libraries
   a) import numpy as np
   b) from sklearn.ensemble import RandomForestClassifier

2) Initialize Random Forest classifier with parameters: n_estimators, random_state.
   a) clf = RandomForestClassifier(n_estimators=n_estimators, random_state=random_state)

3) Set n_estimators and random_state as class attributes.
   a) self.n_estimators = n_estimators
   b) self.random_state = random_state

4) Define a method fit(X, y):
   a) Train the Random Forest classifier on the input data X and labels y using the fit() method.

5) Define a method predict(X):
   a) Predict the labels for the input data X using the predict() method of the Random Forest classifier.
   b) Return the predicted labels.

6) Define a method score(X, y):
   a) Evaluate the accuracy of the Random Forest classifier on the input data X and labels y using the score() method.
   b) Return the accuracy score.

VI. RESULT AND ANALYSIS

The research underscores the critical role of deepfake audio detection in addressing present challenges and shaping future detection methodologies, model architectures, and practical applications. As machine learning algorithms, signal processing techniques, and cybersecurity domains continue evolving, there is an opportunity to enhance the precision, speed, and reliability of deepfake audio detection systems.

VII. APPLICATION

A techno-scientific system containing deepfake audio detection for cyber securities, media forensics, and content moderation shows immense prospects. In cybersecurity sphere, the detection of manipulated audio content can assist in the prevention of fraudulent acts like, 'broken voice skewing' or phishing where social engineering tactics that produce audio content are applied. Hence, a media forensic system can offer the assistance to law enforcement officials and journalists as proof of authenticity of audio records taken in court proceedings or used in journalistic investigations. Furthermore, artifical intelligence can be utilized for content moderation on internet sites and social media networks to tackle the spread of misinformation and hate speech as well as harmful content by marking and deleting deepfake audio content that violates the community guidelines. To summarize, this smart system of detecting deepfake audio addresses many of the hitherto existing serious challenges on safeguarding the integrity and authenticity of the content produced across digital platforms and communication channels.

VIII. CONCLUSION

Deepfake audio detection not only promotes the existing capabilities to combat fake audio, but also enables the future trend in development of detection approaches, architecture of models, and applications. Considering the still-evolving machine learning algorithms, signal processing, and cybersecurity domains, there is a chance for improving the precision, speed, and reliability of deepfake audio detection systems. Future research might focus on exploring different feature extraction methods, including temporal and spatial information, for the purpose of enhancing the discriminative ability of audio-based detection models. In addition, the ongoing work on deep learning techniques, including convolutional neural networks (CNNs) and recurrent neural networks (RNNs), may lead to the emergence of sophisticated detection models that can detect the presence of fine manipulations or even adversarial attacks in audio content. On top of that, deepfake audio detection is likely to spread into fields such as real-time monitoring of audio streams, automated content moderation, and increasing the integrity of voice-based identification systems counters threats.

IX. FUTURE SCOPE

The future scope of deepfake audio detection extends beyond the current capabilities and implementations, offering avenues for advancements and innovations in detection techniques, model architectures, and application domains. With ongoing research and development in machine learning, signal processing, and cybersecurity, there is potential for enhancing the accuracy, efficiency, and robustness of deepfake audio detection systems. Future efforts may focus on exploring novel feature extraction methods, such as incorporating temporal and spatial context information, to improve the discriminative power of audio-based detection models. Additionally, advancements in deep learning techniques, including convolutional neural networks (CNNs) and recurrent neural networks (RNNs), may enable the development of more sophisticated detection models capable of detecting subtle manipulations and adversarial attacks in audio content. Furthermore, the application of deepfake audio detection is poised to expand into emerging domains such as real-time monitoring of audio streams, automated content moderation, and enhancing the resilience of voice-based authentication systems against spoofing attacks.

X. ACKNOWLEDGEMENT

We thank Prof. Jyoti Parashar* our project's mentor, Dr. Akhilesh Das Gupta Institute of Professional Studies. Whose leadership and support have served as the compass guiding us through the challenging terrain of this research. Her valuable feedback and contribution remarkably enhanced our manuscript.

REFERENCES

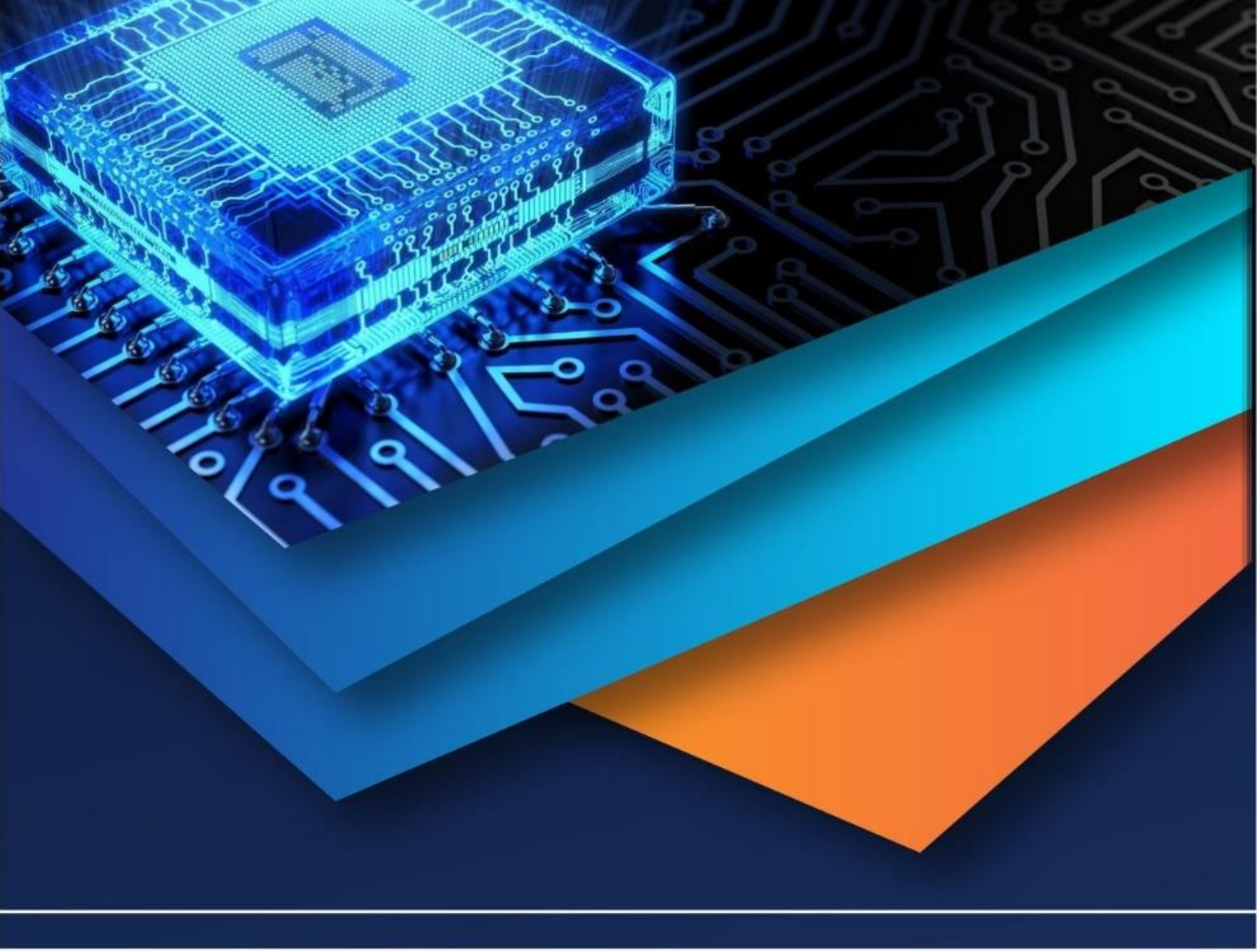
[1] Govindu, Aditi & Kale, Preeti & Hullur, Aamir & Gurav, Atharva & Godse, Parth. (2023). Deepfake audio detection and justification with Explainable Artificial Intelligence (XAI). 10.21203/rs.3.rs-3444277/v1.

[2] Li, Lanting, Tianliang Lu, Xingbang Ma, Mengjiao Yuan, and Da Wan. 2023. "Voice Deepfake Detection Using the Self-Supervised Pre-Training Model HuBERT" *Applied Sciences* 13, no. 14: 8488. https://doi.org/10.3390/app13148488

[3] Gourab Naskar, Sk Mohiuddin, Samir Malakar, Erik Cuevas, Ram Sarkar, Deepfake detection using deep feature stacking and meta-learning, Heliyon, Volume 10, Issue 4, 2024,e25933,ISSN 2405-8440, https://doi.org/10.1016/j.heliyon.2024.e25933.

[4] Almutairi, Zaynab, and Hebah Elgibreen. 2022. "A Review of Modern Audio Deepfake Detection Methods: Challenges and Future Directions" *Algorithms* 15, no. 5: 155. https://doi.org/10.3390/a15050155

[5] Hamza, Ameer & Javed, Abdul Rehman & Iqbal, Farkhud & Kryvinska, Natalia & Almadhor, Ahmad & Jalil, Zunera & Borghol, Rouba. (2022). Deepfake Audio Detection via MFCC Features Using Machine Learning. IEEE Access. PP. 1-1. 10.1109/ACCESS.2022.3231480.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY