

# Convolutional Neural Network-Based Detection and Identification of Deepfake Audio

Oluwasogo Adekunle Okunade

Department of Computer Science, Faculty of Computing, National Open University of Nigeria, Abuja, Nigeria

Corresponding Author's Email: [aokunade@noun.edu.ng](mailto:aokunade@noun.edu.ng)

Received on: August, 2022

Revised and Accepted on: September, 2022

Published on: October, 2022

## ABSTRACT

This study addresses the pressing concern of deepfake audio detection through a comprehensive approach involving convolutional neural networks (CNNs), audio feature extraction, and model evaluation. The research utilises a dataset from Kaggle with labelled deepfake and real audio files and employs the Librosa library for feature extraction, followed by data organisation and preprocessing. Exploratory Data Analysis (EDA) provides insights into class distribution and statistical summaries. Two models, a neural network and LightGBM, are trained and evaluated, with LightGBM exhibiting superior accuracy, precision, recall, and F1-score. The study's implications emphasise the need for ongoing research on emerging deepfake technologies and recommend deploying the robust LightGBM model in real-world scenarios. The study advances understanding of fake audio detection methods, particularly amid the evolving landscape of deepfake techniques, and underscores the importance of adaptable, accurate models for safeguarding against potential threats posed by manipulated audio content.

**Keywords:** Audio feature extraction, Exploratory Data Analysis (EDA), Neural network, LightGBM, Emerging deepfake technologies, Fake audio detection methods, Threats, Manipulated audio content, CNN.

## 1.0 INTRODUCTION

Despite modern internet technology, life transformation still lacks the security necessary to provide users with safe access and defence against harmful attacks. A speech spoofing detection system based on convolutional neural networks and various audio features has been proposed to distinguish human speech from synthetic voices. Worst-case scenarios involve the use of deepfake audio, which can endanger personal property and reputations and threaten the entire nation through unethical use intended to harm third parties (Qais et al., 2022). The deep learning method is built on a convolutional neural network (CNN), a type of black box for detecting audio objects (Kumar et al., 2022). Deepfake voices are an emerging technology that is always changing, there is limited research on the subject, and only a few practical solutions are available. This could lead to cybercrimes such as scams and the exploitation of personal data. The task of determining whether a particular media file is authentic falls within the purview of digital forensics, particularly multimedia forensics (Zawali et al., 2021). The advancement of deep learning techniques, including generative adversarial networks and variational autoencoders, has improved the realism and plausibility of multimedia content. Even if digital data, especially movies captured from a particular perspective, provide an accurate depiction of reality, their authenticity is called into question by the widespread use of content manipulation tools (Dixit et al., 2023).

The analysis process is a key element of digital forensics as an application of deepfake detection, for instance, forensic examiners properly analyze the legitimacy of a given false audio multimedia file, specifically, deep fakes that leverage powerful machine learning algorithms to build a fake audio component; a critical study of the component of such an audio file is required. This study assesses current methods for deep learning-based deepfake audio detection, including feature extraction, data organisation, target labelling, data preprocessing, exploratory data analysis, and model training. The use of both a neural network and LightGBM model allows for a comprehensive evaluation, with a Streamlit application designed for practical deployment, offering users an intuitive interface for deepfake prediction.

## 2.0 LITERATURE REVIEW

Several studies have presented a wide range of methods for identifying deepfakes in images and videos. As technology has advanced, experts have begun to look at audio deepfakes and how to spot them. This study aims to provide a summary of pertinent research in this domain, as there is currently no comprehensive overview of audio deepfake production and detection algorithms (Dixit et al., 2023). Numerous detection techniques have been developed to distinguish between actual speech and phoney audio recordings in the case of deepfakes. Many machine learning and deep learning models have been created that employ various techniques to identify false audio. The general method of Audio detection (AD), as depicted in Figure 1, is explained by the following

strategies. Each audio clip should first undergo preprocessing to extract appropriate audio features, such as Mel-spectrograms (Almutairi & Elgibreen, 2022).

The concept of detecting forged media is a subject being broadly researched (Zheng, Zhang, & Thing, 2019). Two major detection categories are artefact-specific methods and other specialised approaches.

#### Artefact-specific Detection of Deep Fakes

Artefacts are often generated by deepfake models, which may not be detectable by human observation. However, artefacts can be easily identified by machine learning models. Forensic analysis can also be used to detect artefacts.

Some artefacts become apparent when the generated content is merged into the frames. Researchers suggested frequency analysis, quality measures and edge detector (Durall et al., 2019; Zhang et al., 2017). Li et al. (2020) used an explicit method to detect boundaries: a CNN network was trained to predict the blending boundary of an image with real or fake labels. The training was done on a dataset of face swaps created by combining similar faces identified by matching facial landmarks. This approach gives the model the benefit of focusing on the blending boundary rather than on the artefacts that generative models create.

A fake face can appear abnormal when compared to the remaining frames. The residuals in lighting and face warping could be an indication that the media was generated. Li et al. (2020) used an approach in which they created a contrast between a generated foreground and an unaltered background; this was done with a patch-and-pair CNN. The context and the face were encoded by Nirkin et al. (2020) using an encoder-decoder, and the difference between the encoding and the encoded image was then passed to a classifier.

Deepfakes can also be analysed using latent patterns and features that the model imprinted on the generated media. It was identified by Marra et al. (2019) and Yu (2019) that GAN-generated content leaves distinct imprints; this indicates the possibility of classifying the generator if the content is given, even when noise and compression are present. Content related to imperfections was examined by Yang et al. (2019). Instead of looking for residuals and imprints, they found that head poses appear inconsistent in deepfake videos. Hence, they were able to detect deepfakes by predicting facial landmarks. Another strategy was employed by Wang et al. (2020), who built classifier models that focus on observing imperfections rather than residuals. They were able to do this using a dataset obtained from ProGAN, as its images have a much lower frequency of artefacts.

From another perspective, anomalies could be identified by observing and monitoring mannerisms and other related behaviours, based on the large volume of data on the target. For instance, Mittal (2020) demonstrated that behaviours can be inferred without reference to the target's footage, using a customised Siamese network to detect emotions in audio and video and to distinguish real from fake videos. Thus, this technique captures inconsistencies in the emotion deduced from the audio clip and video content.

It was suggested by Conotter et al. (2014) that generated multimedia files will be devoid of physiological stamps, and they then identified synthetically generated faces by inspecting their heart rate. The technique of closely monitoring eye-blink patterns, whether irregular or not, was employed by Li et al. (2018) and Jung (2020), who used pulse signals to determine the specific model used to produce deepfake media.

Korshunov & Marcel (2018) found that dubbed videos can be detected by observing the correlation between speech and mouth-region landmarks. An approach to identifying when mouth shapes are inconsistent with the utterance was considered by Agarwal et al. (2020). Therefore, they zoomed in on the sounds where the mouth is expected to be closed, since deepfakes generally cannot create consistent mouth shapes throughout.

Furthermore, achieving realistic temporal coherence is often difficult; however, it is feasible to focus on the resulting effects to detect fake videos. RNNs were used by Güera & Delp (2018) to detect jitters and flickers, and by Sabir et al. (2019) to detect deepfake multimedia using an LSTM on the face region.

Instead of fixing the focus on a particular artefact, certain scholars trained deep neural networks to serve as generic classifiers, allowing the networks to determine which features to evaluate. These non-artefact detection modes make use of either classification or anomaly detection. It has been shown by Marra et al. (2018), Nguyen et al. (2019), and Rossler et al. (2019) that deep neural networks are likely to outperform traditional image forensic tools when evaluating images. Researchers demonstrated the effective detection of deepfake videos using CNN architectures (Abdul et al., 2024; Ding et al., 2020; Tariq et al., 2018). In their study, Fernando et al. (2019) found that certain deep fakes are detected by CNNs only when trained on those specific deep fakes; they may not detect a type of deep fake they were not trained on. The authors therefore proposed the hierarchical Memory Network (HMN) model, which evaluates the constituents of a face and the faces it has seen previously. This model encodes the facial region

and then processes it with a bi-directional GRU. The memory module receives the final encodings, compares them with recently viewed encodings, and then makes its prediction. Varieties of spatio-temporal networks and feature extractors to be used as the model for detecting temporal deepfakes were sampled by De et al. (2020). They discovered that a 3D CNN that simultaneously views multiple frames outperformed RNN and ID3 architectures. Even though classifiers have been used to detect deepfakes, they appear to be quite limited, as deepfake content creators can evade detection by using adversarial machine learning (Mirsky & Lee, 2021).

Anomaly detection models can also be used to detect deepfake media. These models are trained on a normal dataset and then left to detect outliers as they view streams of data. Implicatively, this method makes no assumptions about how a deepfake attack appears. Therefore, it is capable of spotting deep fake multimedia that has been created with unknown methods, it is also

able to generalize what a deep fake could be in a better way. Waleed et al. (2022) employed anomaly detection by evaluating the coverage of a face recognition model. In this technique, the model overcame image noise by extracting a clearer signal than could be obtained from raw pixels alone. A one-class VAE was applied by (Khalid, & Woo, 2020) for the purpose of reconstructing real images. So, an anomaly score is calculated by evaluating the mean squared error (MSE) between the mean constituent of the encoded image and that of the reconstructed image.

The detection model receives these features and uses them to carry out the necessary functions, such as training. If the job involves nonlinearity, the output can be fed into any fully connected layer with an activation function to obtain a predicted probability that class 0 is fake and class 1 is real. However, there is a cost associated with accuracy (Almutairi & Elgibreen, 2022). The AD detection process is depicted in Figure 1.

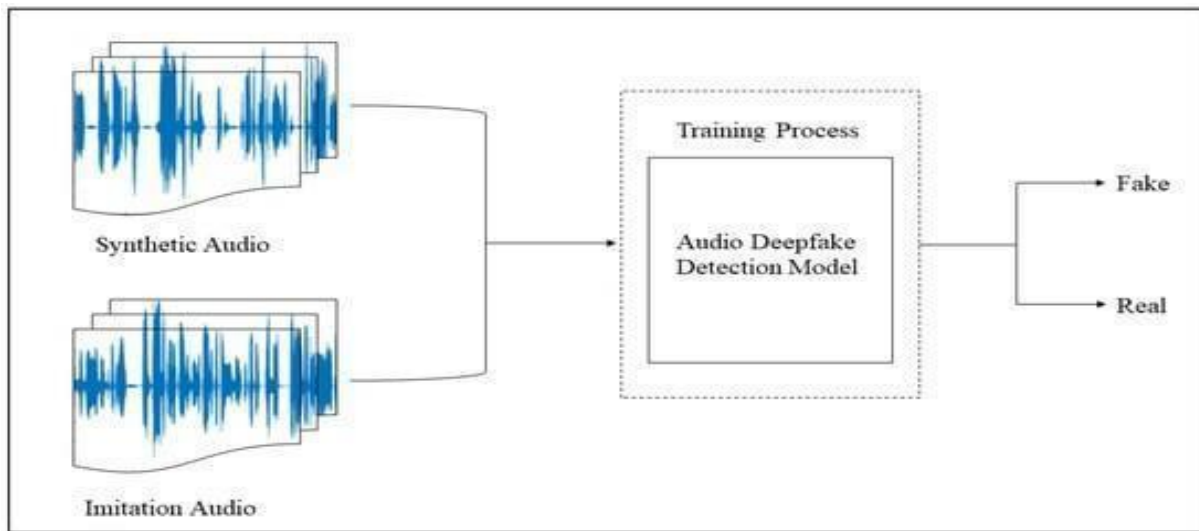


Figure 1: An illustration of the AD detection process

### 2.1 Fake Audio Detection Methods

For AD detection, classical machine learning models have been frequently used. In two ways, (Rodríguez-Ortega, Ballesteros, & Renza., 2020.) added to the body of knowledge on the detection of phoney audio. Using the imitation method, they first created a synthetic audio dataset by extracting entropy features from both false and genuine audio. The researchers developed an ML model using Logistic Regression (LR) to identify false audio, based on the produced H-Voice dataset. The model's task detection success rate was 98%; however, manual pre-processing of the data was required to extract the pertinent characteristics.

A Quadratic Support Vector Machine (Q-SVM) model was proposed in (Singh & Singh, 2021) to distinguish artificial from human speech. The authors split the audio into two classes, human and AI-generated, after deciding to use the binary classification methodology. Several Machine learning techniques, including boosted tree ensembles, linear discriminant analysis, quadratic discriminant analysis, linear SVM, weighted K-Nearest Neighbours (KNN), and LR, were compared with this model. They discovered that Q-SVM outperformed other traditional approaches by 97.56%, with a misclassification rate of 2.43%. The studies described

thus far suggest that, to ensure good performance, extensive preprocessing must be performed before training, and that features for the ML models must be manually extracted. However, because this is time-consuming and can yield inconsistent results, researchers have developed high-level DL algorithms.

In response, the CNN model in (Subramani & Rao (2020)) developed a novel method using two CNN models, RES-EfficientCNN and EfficientCNN, for detecting synthetic audio. Consequently, when tested on the ASV spoof challenge 2019 dataset, RES-EfficientCNN achieved a higher F1-score of 97.61 than EfficientCNN (94.14). To distinguish between synthetic and imitation sounds, Ballesteros et al. (2021) created a classification model called Deep4SNet that displayed the audio dataset using a 2D CNN model (histogram). Deep4SNet demonstrated a 98.5% detection accuracy for synthetic and imitation sounds. However, Deep4SNet's performance was not scalable, and the data transformation procedure affected it.

The Arabic Diversified Audio (AR-DAD) dataset of Quranic audio samples was the target of the suggested method's imitation-based fakeness (Lataifeh & Elnagar, 2020). They evaluated the capacity of CNNs and BiLSTMs to discriminate between authentic and fake voices. Furthermore, tests were conducted on machine learning techniques, including SVM, SVM-Linear, Radial Basis Function (SVMRBF), LR, Decision Tree (DT), RF, and Gradient Boosting (XGBoost). The proposed imitation-based fakeness approach was used on the Arabic Diversified Audio (AR-DAD) dataset of Quranic audio samples (Lataifeh & Elnagar, 2020). They assessed the ability of CNNs and BiLSTMs to distinguish between real and artificial voices. Additionally, experiments were run using machine learning methods such as SVM, SVM-Linear, LR, Decision Tree (DT), RF, Gradient Boosting (XGBoost), and Radial Basis Function (SVMRBF).

Various fake audio detection methods employ classical machine learning (e.g., Logistic Regression, Quadratic SVM) and deep learning models (e.g., CNNs) to discern genuine from manipulated audio. Classical approaches require manual preprocessing, achieving high success rates but with data limitations. Deep learning models, such as RES-EfficientCNN and Deep4SNet, achieve promising results for synthetic audio detection without extensive manual feature extraction. Challenges include scalability issues and the impact of data transformation. These methods collectively advance fake audio detection, addressing the growing sophistication of deepfake audio techniques.

### 3.0 MATERIALS AND METHODS

#### 3.1 Audio Dataset Overview

The audio dataset utilized in this study is sourced from Kaggle, a well-known platform for data science competitions and datasets. The dataset comprises two main directories: FAKE\_AUDIO\_DIR and REAL\_AUDIO\_DIR. The audio files in these directories are integral to the investigation, and each is analysed for deepfake classification.

Source: The dataset is retrieved from Kaggle, a platform that hosts diverse datasets contributed by the community.

Directories:

FAKE\_AUDIO\_DIR: Contains audio files associated with deepfake content.

REAL\_AUDIO\_DIR: Encompasses audio files representing genuine, non-fabricated content.

File Types:

Audio files with extensions .wav and .mp3 are considered for analysis.

Feature Extraction Process

The extraction of relevant features from audio data is a crucial step in developing a robust classification model. The Librosa library in Python is employed for its effectiveness in extracting various audio features.

Librosa Library: Librosa is a Python package designed for music and audio analysis, providing tools for feature extraction, visualisation, and manipulation.

Feature Set: The following audio features are extracted from each audio file in the dataset:

Chroma Short-Time Fourier Transform (chroma\_stft), Root Mean Square Error (rmse), Spectral Centroid, Spectral Bandwidth, Spectral Rolloff, Zero Crossing Rate, Spectral Flatness, Delta Chroma, Tonnetz.

Data Organisation: Extracted features are organized into a structured data frame named data for systematic analysis.

Target Labelling: Each audio file is labelled based on its origin, distinguishing between deepfake and real content. This step is fundamental for supervised machine learning, where the model learns patterns associated with each class.

Class Labels:

Class 0: Deepfake (Files from FAKE\_AUDIO\_DIR)

Class 1: Real (Files from REAL\_AUDIO\_DIR)

Data Preprocessing: The labelled features are prepared for model training by creating training and testing datasets with an 80-20 split.

#### 3.2 Exploratory Data Analysis (EDA)

The statistical summary provides insights into the central tendencies and spread of the audio features. Key metrics such as mean, standard deviation, minimum, maximum,

and quartiles are summarized for each feature. This information helps in understanding the overall distribution of the dataset.

**Summary Statistics:** The statistical summary provides a comprehensive overview of key descriptive statistics for

each feature in the dataset. This includes measures such as mean, standard deviation, minimum, maximum, and quartiles. Understanding these statistics is fundamental to grasping the central tendencies and variability in the audio features. Presented in Table 1 is the summary of statistics.

Table 1: Summary Statistics

|                    | count | mean     | std       | min       | 25%       | 50%       | 75%       | max       |
|--------------------|-------|----------|-----------|-----------|-----------|-----------|-----------|-----------|
| chroma_stft        | 100   | 0.376685 | 0.036972  | 0.290720  | 0.355155  | 0.373720  | 0.398778  | 0.484781  |
| rmse               | 100   | 0.072372 | 0.037461  | 0.020179  | 0.029779  | 0.078536  | 0.105657  | 0.132745  |
| spectral_centroid  | 100   | 1680.295 | 159.61971 | 1307.6559 | 1557.4154 | 1689.0022 | 1807.5795 | 1987.5927 |
| spectral_bandwidth | 100   | 1666.929 | 108.84131 | 1437.0069 | 1593.0952 | 1666.4373 | 1749.1852 | 1946.3547 |
| spectral_rolloff   | 100   | 3259.713 | 330.99378 | 2560.0780 | 3007.4353 | 3280.5410 | 3527.3297 | 4006.5746 |
| zero_crossing_rate | 100   | 0.088668 | 0.012400  | 0.061416  | 0.080586  | 0.087880  | 0.097988  | 0.120470  |
| spectral_flatness  | 100   | 0.007714 | 0.003886  | 0.000989  | 0.005338  | 0.007091  | 0.009653  | 0.023121  |
| delta_chroma       | 100   | -        | 0.000793  | -0.001999 | -0.000820 | -0.000480 | 0.000265  | 0.001981  |
| tonnetz            | 100   | -        | 0.007042  | -0.020903 | -0.005410 | -0.002596 | 0.002181  | 0.013602  |
| target             | 100   | 0.500000 | 0.502519  | 0.000000  | 0.000000  | 0.500000  | 1.000000  | 1.000000  |

**Visualizations:** Visualization techniques play a pivotal role in uncovering patterns and trends within the data. This section employs various graphical representations, such as histograms, scatter plots, and pair plots, to showcase the distribution and relationships among

different audio features. Visualization aids in identifying potential outliers, clusters, or any discernible patterns that may influence the classification task. Histograms of Audio Features is depicted in Figure 2.

Histograms of Audio Features

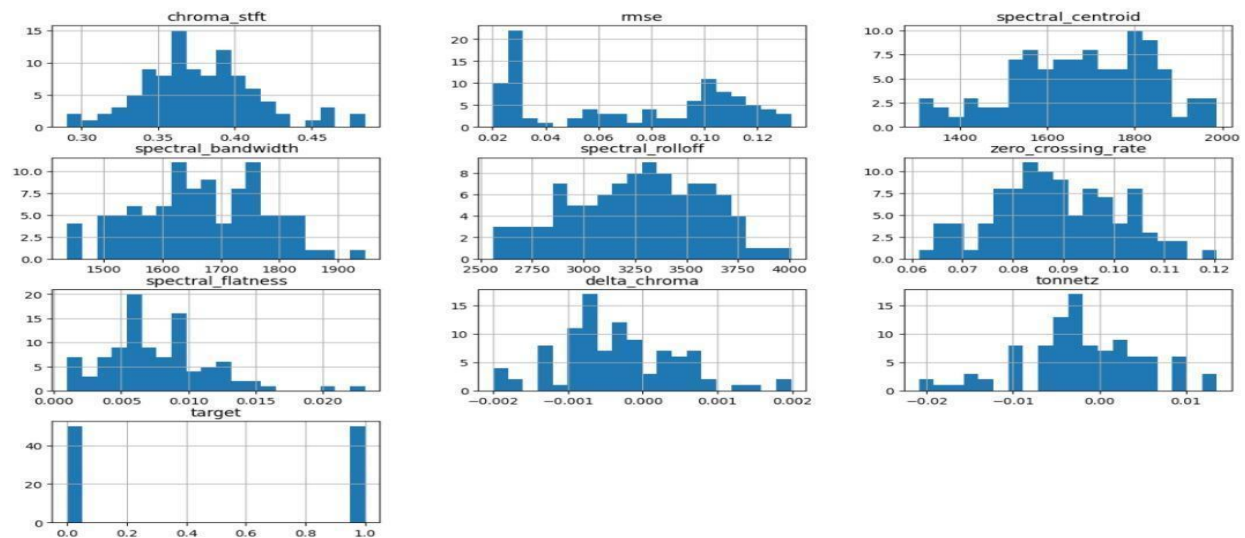


Figure 2: Histogram of Audio Features

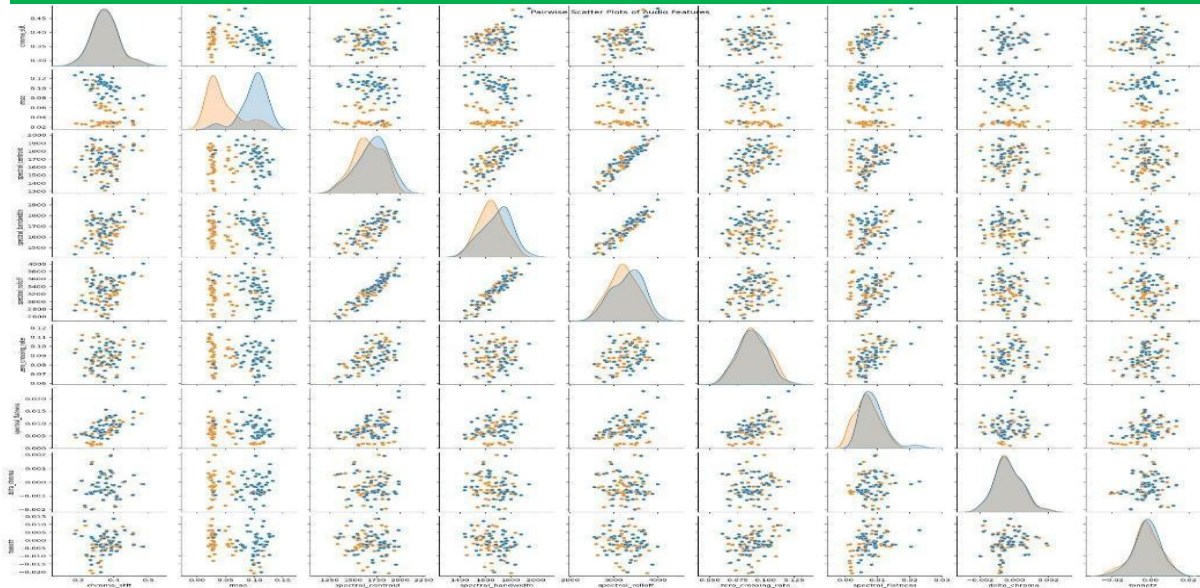


Figure 3: Pairwise Box Plot

Depicted in Figure 4 is the Box Plot of Chroma Short-Time Fourier Transform (chroma\_stft) by Class.

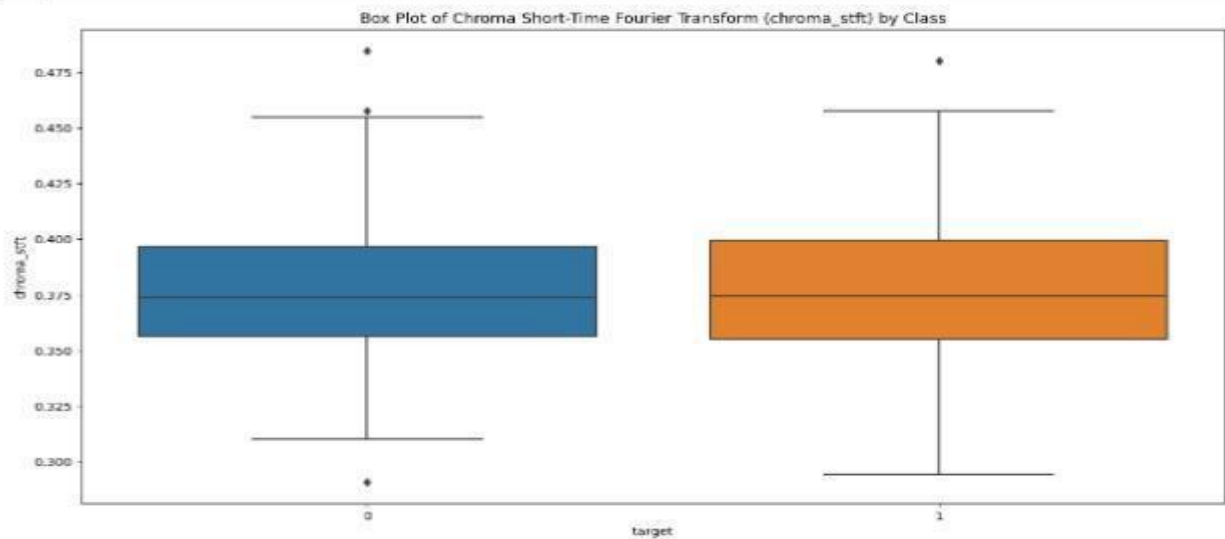


Figure 4: Box Plot of Chroma Short-Time Fourier Transform (chroma\_stft) by Class

### 3.3 Class Distribution Analysis

Analyzing the distribution of classes is essential for understanding the balance or imbalance in the dataset. The following visualization provides insights into the distribution of deepfake (0) and real (1) audio samples:

Figure 5 presents the class distribution: Deepfake (0) vs Real (1).

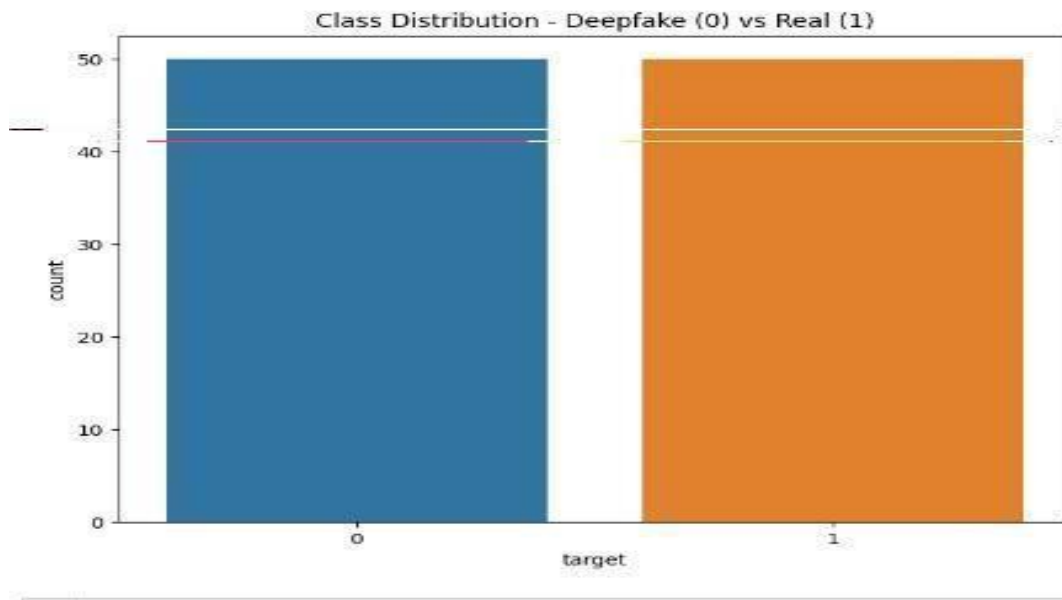


Figure 5: Deep fake vs Real fake

## 4.0 RESULT

### 4.1 Overview of Extracted Features

The extracted audio features play a crucial role in understanding the characteristics of the dataset. The following audio features are extracted from each audio file in the dataset: Chroma Short-Time Fourier Transform (chroma\_stft), Root Mean Square Error (rmse), Spectral Centroid, Spectral Bandwidth, Spectral Rolloff, Zero Crossing Rate, Spectral Flatness, Delta Chroma, Tonnetz.

### 4.2 Train-Test Split

To assess the model's performance, the dataset is divided into training and testing sets. The **train\_test\_split**

function from scikit-learn is used for this purpose. The split is stratified, preserving the class distribution in both the training and testing sets.

### 4.3 Model Training

#### 4.3.1 Neural Network Model (Architecture)

The neural network model architecture is designed for audio classification. It consists of multiple layers, including Dense, Dropout, and BatchNormalization layers. The model is compiled with the Adam optimizer and binary cross-entropy loss function. Figure 6 represents a summary of the neural network architecture.

Model: "sequential"

| Layer (type)                              | Output Shape | Param # |
|-------------------------------------------|--------------|---------|
| dense (Dense)                             | (None, 128)  | 1280    |
| dropout (Dropout)                         | (None, 128)  | 0       |
| dense_1 (Dense)                           | (None, 32)   | 4128    |
| batch_normalization (Batch Normalization) | (None, 32)   | 128     |
| dropout_1 (Dropout)                       | (None, 32)   | 0       |
| dense_2 (Dense)                           | (None, 1)    | 33      |
| =====                                     |              |         |
| Total params: 5569 (21.75 KB)             |              |         |
| Trainable params: 5505 (21.50 KB)         |              |         |
| Non-trainable params: 64 (256.00 Byte)    |              |         |

Figure 6: Model Architecture Summary

## 4.4 Performance Evaluation

## 4.4.1 Model Evaluation

Both the neural network and LightGBM models are evaluated using the test set. The evaluation metrics

provide insights into the models' performance, including accuracy, precision, recall, and F1 score. Figure 7 is the accuracy score of LGBM Model.

ACCURACY SCORE WITH LGBM: 0.95

```
( '          precision    recall  f1-score   support\n
 '          0          1.00      0.90      0.95         10\n
 '          1          0.91      1.00      0.95         10\n
 '\n'
 '      accuracy                   0.95         20\n
 '      macro avg          0.95      0.95      0.95         20\n
 ' weighted avg          0.95      0.95      0.95         20\n')
```

Figure 7: Accuracy Score with LGBM

**Accuracy and Class Metrics:** The LightGBM model achieves an impressive accuracy of 95%, indicating its proficiency in correctly classifying audio samples. Both precision and recall for both classes are notably high, resulting in balanced F1-scores. Precision reflects the model's ability to avoid misclassifying real audio samples as deepfake and vice versa. Recall demonstrates the model's capability to identify a high proportion of true positive cases.

**Strengths:** The model excels in correctly identifying both deepfake and real audio samples, with high precision and recall for both classes. The balanced metrics indicate a robust performance across both classes, suggesting that the model is effective in discriminating between deepfake and real audio.

**Neural Network Model Performance**

Depicted in Figure 8 is the performance of Neural Network for detecting audio fake.

ACCURACY SCORE WITH NEURAL NET: 0.65

```
( '          precision    recall  f1-score   support\n
 '          0          0.80      0.40      0.53         10\n
 '          1          0.60      0.90      0.72         10\n
 '\n'
 '      accuracy                   0.65         20\n
 '      macro avg          0.70      0.65      0.63         20\n
 ' weighted avg          0.70      0.65      0.63         20\n')
```

Figure 8: Neural Network Model Performance

**Accuracy and Class Metrics:** The neural network model achieves 65% accuracy, indicating lower overall performance compared to LightGBM. The class metrics indicate imbalances, with significantly lower precision for class 0 (deepfake) and lower recall for class 1 (real).

**Areas for Improvement:**

The neural network struggles with misclassifications, as evidenced by lower precision on deepfake samples and lower recall on real samples. The model's performance suggests a potential need for fine-tuning or adjustments to improve its ability to correctly classify both classes.

Table 2: Model Comparison and Interpretation

| Metric                 | LightGBM | Neural Network |
|------------------------|----------|----------------|
| Accuracy               | 0.95     | 0.65           |
| Precision (Class 0)    | 1.00     | 0.80           |
| Precision (Class 1)    | 0.91     | 0.60           |
| Recall (Class 0)       | 0.90     | 0.40           |
| Recall (Class 1)       | 1.00     | 0.90           |
| F1-Score (Class 0)     | 0.95     | 0.53           |
| F1-Score (Class 1)     | 0.95     | 0.72           |
| Support (Class 0)      | 10       | 10             |
| Support (Class 1)      | 10       | 10             |
| Macro Avg Precision    | 0.95     | 0.70           |
| Macro Avg Recall       | 0.95     | 0.65           |
| Macro Avg F1-Score     | 0.95     | 0.63           |
| Weighted Avg Precision | 0.95     | 0.65           |
| Weighted Avg Recall    | 0.95     | 0.65           |
| Weighted Avg F1-Score  | 0.95     | 0.63           |

- 1. Superiority of LightGBM:** The LightGBM model outperforms the neural network across all key metrics as observed in table 2 above, indicating its suitability for this audio classification task. The balanced precision and recall for both classes contribute to the model's robust performance.
- 2. Recommendations:** Further investigation is recommended to understand the specific characteristics of the audio features and their impact on model performance. Hyperparameter tuning and feature engineering may enhance the neural network's performance, aligning it more closely with the LightGBM model.
- 3. Consideration for Deployment:** Given the superior performance of the LightGBM model, it may be the preferred choice for deployment in real-world scenarios. Continued monitoring and evaluation are crucial, as the model's performance may be influenced by changes in the distribution of incoming data.

#### 4.5 Deployment

##### 4.5.1 Streamlit Application Overview

###### File Uploader:

Users can upload video, audio, or image files for analysis. Accepted video formats include mp4, avi, mkv, and x-matroska. Accepted image formats include jpg, jpeg, and png. Accepted audio formats are processed through audio feature extraction.

###### Predict Button:

Users trigger the prediction by clicking the "Predict" button. The application automatically identifies the type of file (Video, Image, or Audio).

##### 4.5.2 File Type Recognition and Prediction

###### Audio File:

Audio files undergo feature extraction to obtain relevant audio features.

The extracted features are fed into a LightGBM classifier for deepfake prediction.

The output screen is shown in Figures 9 and 10 below.

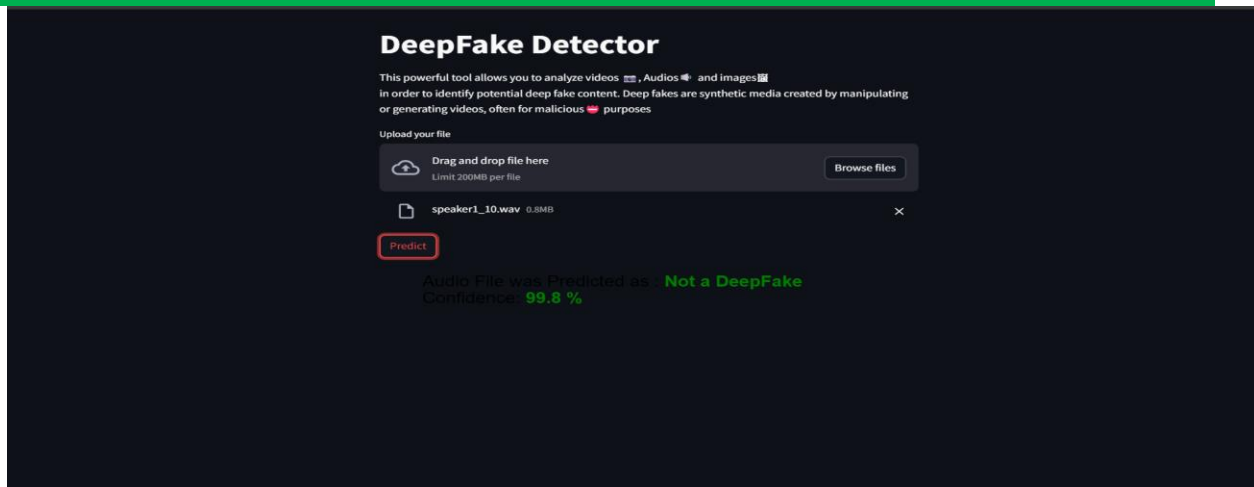


Figure 9: DeepFake Detector Result (a)

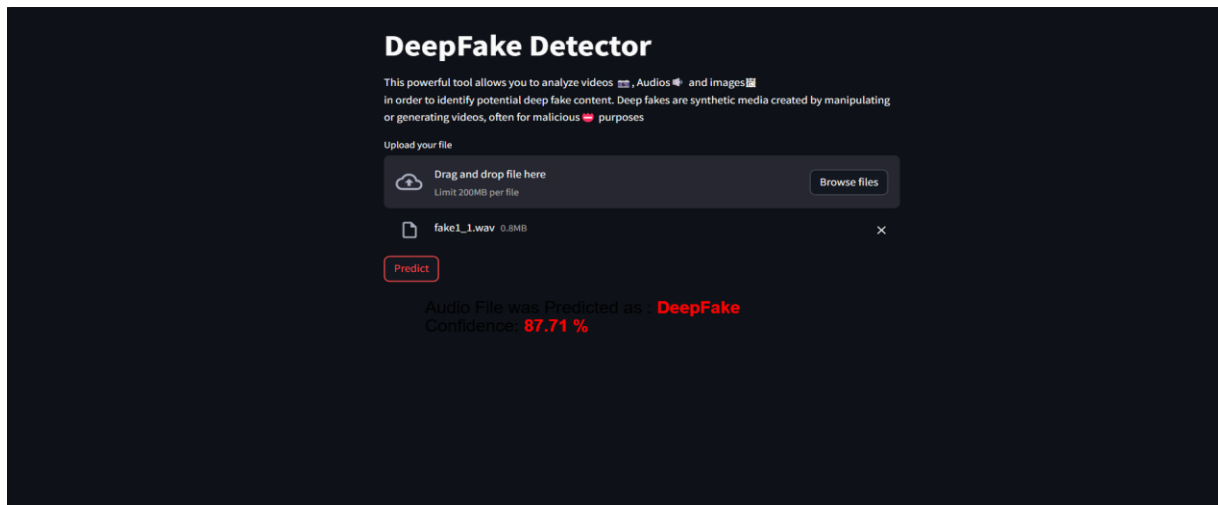


Figure 10: DeepFake Detector Result (b)

The application displays the prediction result, indicating whether the file is classified as a deepfake or not. Confidence in the prediction is presented as a percentage. The result is visually highlighted with different colours ("red" for deepfake, "green" for not a deepfake).

## 5.0 CONCLUSION

The study addresses the pressing issue of deepfake audio detection, recognising the potential threats posed by synthetic voices in various domains. Leveraging convolutional neural networks (CNNs) and a diverse set of audio features, the research explores methods to distinguish genuine from manipulated audio files. Through a comprehensive evaluation that includes statistical summaries, visualisations, and model training with both neural networks and LightGBM, the study provides insights into the performance of these techniques. Notably, the LightGBM model demonstrates

superior accuracy, precision, and recall, showcasing its effectiveness in identifying deepfake audio.

## 6.0 FUTURE WORK

Future work includes ongoing research into evolving deepfake technologies, exploration of advanced deep learning techniques, and fine-tuning the neural network model to improve performance. The superior LightGBM model is suggested for real-world deployment, with an emphasis on continuous monitoring and adaptation to changing data distributions. Additionally, the development of user-friendly tools that incorporate feedback and updates is crucial for effective practical applications of deepfake detection methods.

## References

- Abdul Qadir, Rabbia Mahum, Mohammed A. El-Meligy, Adham E. Ragab, Abdulmalik AlSalman, M. A. (2024). An efficient deepfake video detection using robust deep learning, *Heliyon*, Volume 10, Issue 5, e25757, ISSN 2405-8440, <https://doi.org/10.1016/j.heliyon.2024.e25757>.
- Agarwal, S., Farid, H., Fried, O., & Agrawala, M. (2020). Detecting deep-fake videos from phoneme-viseme mismatches. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops* (pp. 660-661).
- Almutairi, Z., & Elgibreen, H. (2022). A review of modern audio deepfake detection methods: Challenges and future directions. *Algorithms*, 15(5), 155. <https://doi.org/10.3390/a15050155>
- Ballesteros, D. M., Rodriguez-Ortega, Y., Renza, D., & Arce, G. (2021). Deep4SNet: Deep elearning for fake speech classification. *Expert Systems with Applications*, 184, 115465. <https://doi.org/10.1016/j.eswa.2021.115465>
- Conotter, V., Bodnari, E., Boato, G., & Farid, H. (2014, October). Physiologically-based detection of computer-generated faces in video. In *2014 IEEE International Conference on Image Processing (ICIP)* (pp. 248-252). IEEE.
- Ding, X., Raziei, Z., Larson, E. C., Olinick, E. V., Krueger, P., & Hahsler, M. (2020). Swapped face detection using deep learning and subjective assessment. *EURASIP Journal on Information Security*, 2020(1), 1-12.
- Dixit, A., Kaur, N., & Kingra, S. (2023). Review of audio deepfake detection techniques: Issues and prospects. *Expert Systems*, 40(8). <https://doi.org/10.1111/exsy.13322>
- Do, N. T., Na, I. S., & Kim, S. H. (2018). Forensics face detection from GANs using convolutional neural network. *ISITC*, 2018, 376-379.
- Durall, R., Keuper, M., Pfrendt, F. J., & Keuper, J. (2019). Unmasking deepfakes with simple features. *arXiv preprint arXiv:1911.00686*.
- Fernando, T., Fookes, C., Denman, S., & Sridharan, S. (2019). Exploiting human social cognition for the detection of fake and fraudulent faces via memory networks. *arXiv preprint arXiv:1911.07844*.
- Güera, D., & Delp, E. J. (2018, November). Deepfake video detection using recurrent neural networks. In *2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)* (pp. 1-6). IEEE.
- Jung, TackHyun & Kim, Sangwon & Kim, Keecheon. (2020). DeepVision: Deepfakes Detection Using Human Eye Blinking Pattern. *IEEE Access*. PP. 1-1. 10.1109/ACCESS.2020.2988660.
- Khalid, H., & Woo, S. S. (2020). OC-FakeDect: Classifying deepfakes using one-class variational autoencoder. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops* (pp. 656-657).
- Korshunov, P., & Marcel, S. (2018). Deepfakes: A new threat to face recognition? Assessment and detection. *arXiv preprint arXiv:1812.08685*.
- Kumar, B., & Alraisi, S. R. (2022). Deepfakes audio detection techniques using deep convolutional neural network. *2022 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COM-IT-CON)*, 463–468. <https://doi.org/10.1109/COM-IT-CON54601.2022.9850771>
- Lataifeh, M., & Elnagar, A. (2020). Ar-DAD: Arabic diversified audio dataset. *Data in Brief*, 33, 106503. <https://doi.org/10.1016/j.dib.2020.106503>
- Li, Y., Chang, M. C., & Lyu, S. (2018, December). In ictu oculi: Exposing AI created fake videos by detecting eye blinking. In *2018 IEEE International Workshop on Information Forensics and Security (WIFS)* (pp. 1-7). IEEE.
- Li, X., Yu, K., Ji, S., Wang, Y., Wu, C., & Xue, H. (2020, April). Fighting against deepfake: Patch&pair convolutional neural networks (PPCNN). In *Companion Proceedings of the Web Conference 2020* (pp. 88-89).
- Marra, F., Gragnaniello, D., Cozzolino, D., & Verdoliva, L. (2018, April). Detection of GAN-generated fake images over social networks. In *2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)* (pp. 384-389). IEEE.
- Marra, F., Gragnaniello, D., Verdoliva, L., & Poggi, G. (2019, March). Do GANs leave artificial fingerprints? In *2019 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)* (pp. 506-511). IEEE.

- Mirsky, Y., & Lee, W. (2021). The creation and detection of deepfakes: A survey. *ACM Computing Surveys (CSUR)*, 54(1), 1-41.
- Mittal, T., Bhattacharya, U., Chandra, R., Bera, A., & Manocha, D. (2020, October). Emotions don't lie: An audio-visual deepfake detection method using affective cues. In *Proceedings of the 28th ACM International Conference on Multimedia* (pp. 2823-2832).
- Nguyen, H. H., Yamagishi, J., & Echizen, I. (2019, May). Capsule-forensics: Using capsule networks to detect forged images and videos. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 2307-2311). IEEE.
- Nirkin, Y., Wolf, L., Keller, Y., & Hassner, T. (2020). Deepfake detection based on the discrepancy between the face and its context. *arXiv preprint arXiv:2008.12262*.
- Qais, A., Rastogi, A., Saxena, A., Rana, A., & Sinha, D. (2022). Deepfake audio detection with neural networks using audio features. *2022 International Conference on Intelligent Controller and Computing for Smart Power (ICICCSP)*, 1-6. <https://doi.org/10.1109/ICICCSP53532.2022.9862519>
- Rodríguez-Ortega, Y., Ballesteros, D. M., & Renza, D. (2020). A machine learning model to detect fake voice (pp. 3-13). [https://doi.org/10.1007/978-3-030-61702-8\\_1](https://doi.org/10.1007/978-3-030-61702-8_1)
- Rossler, A., Cozzolino, D., Verdoliva, L., Riess, C., Thies, J., & Nießner, M. (2019). Faceforensics++: Learning to detect manipulated facial images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 1-11).
- Sabir, E., Cheng, J., Jaiswal, A., AbdAlmageed, W., Masi, I., & Natarajan, P. (2019). Recurrent convolutional strategies for face manipulation detection in videos. *Interfaces (GUI)*, 3(1), 80-87.
- Singh, A. K., & Singh, P. (2021). Detection of AI-synthesized speech using cepstral & bispectral statistics. *2021 IEEE 4th International Conference on Multimedia Information Processing and Retrieval (MIPR)*, 412-417. <https://doi.org/10.1109/MIPR51284.2021.00076>
- Subramani, N., & Rao, D. (2020). Learning efficient representations for fake speech detection. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04), 5859-5866. <https://doi.org/10.1609/aaai.v34i04.6044>
- Tariq, S., Lee, S., Kim, H., Shin, Y., & Woo, S. S. (2018, January). Detecting both machine and human created fake face images in the wild. In *Proceedings of the 2nd International Workshop on Multimedia Privacy and Security* (pp. 81-87).
- Waleed Hilal, S. Andrew Gadsden, John Yawney, (2022). Financial Fraud: A Review of Anomaly Detection Techniques and Recent Advances, *Expert Systems with Applications*, Volume 193, 116429, ISSN 0957-4174, <https://doi.org/10.1016/j.eswa.2021.116429>.
- Wang, S. Y., Wang, O., Zhang, R., Owens, A., & Efros, A. A. (2020). CNN-generated images are surprisingly easy to spot... for now. In *Proceedings of the IEEE*
- Yang, X., Li, Y., & Lyu, S. (2019, May). Exposing deep fakes using inconsistent head poses. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 8261-8265). IEEE.
- Yu, N., Davis, L. S., & Fritz, M. (2019). Attributing fake images to GANs: Learning and analyzing GAN fingerprints. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 7556-7566). /CVF Conference on Computer Vision and Pattern Recognition (pp. 8695-8704).
- Zawali, B., Ikuesan, R. A., Kebande, V. R., Furnell, S., & A-Dhaqm, A. (2021). Realising a push button modality for video-based forensics. *Infrastructures*, 6(4), 54. <https://doi.org/10.3390/infrastructures6040054>
- Zhang, Y., Zheng, L., & Thing, V. L. (2017, August). Automated face swapping and its detection. In *2017 IEEE 2nd International Conference on Signal and Image Processing (ICSIP)* (pp. 15-19). IEEE.
- Zhang, Y., Zheng, L., & Thing, V. L. (2017, August). Automated face swapping and its detection. In *2017 IEEE 2nd International Conference on Signal and Image Processing (ICSIP)* (pp. 15-19). IEEE.
- Zheng, L., Zhang, Y., & Thing, V. L. (2019). A survey on image tampering and its detection in real-world photos. *Journal of Visual Communication and Image Representation*, 58, 380-399.