Al-Qadisiyah JournalFor Engineering Sciences 19 (2026) 270 – 277

Contents lists available at: http://qu.edu.iq

Al-Qadisiyah Journal for Engineering Sciences

Journal homepage: https://qjes.qu.edu.iq

Research Paper

Check for updates

Detecting fake audio using convolutional neural networks for reducing misinformation

Esraa Y. Tarkan [1], ***Mohammed M. Msallam*** [2]✉, ***and Sarah K. Salim*** [3]

[1]*Ministry of Higher Education and Scientific Research, Baghdad, Iraq.*
[2]*College of Artificial Intelligence Engineering, University of Technology, Baghdad, Iraq.*
[3]*Electrical Engineering Department, University of Misan,Misan, Iraq.*

ARTICLE INFO

Article history:
Received 24 June 2025
Received in revised form 12 September 2025
Accepted 13 May 2026

keyword:
Convolutional neural network
Mel Frequency cepstral coefficient
Derivative
Fake audio
Misinformation

ABSTRACT

Recently, a proliferation of techniques capable of replicating sounds has emerged, encompassing manipulated recordings, synthesized audio, and deepfake technologies. These advanced methods for generating artificial sounds have inadvertently given rise to a multitude of issues, prominently including the dissemination of misinformation, propaganda, and significant reputational damage. This article addresses this critical problem by proposing the application of a Convolutional Neural Network (CNN) model designed to predict the authenticity of a given sound. For feature extraction from the audio, Mel Frequency Cepstral Coefficients (MFCC) are utilized. To rigorously assess the robustness of the proposed model, the intricate combinations of these extracted features were further analyzed using derivatives of the MFCC features. The Fake-or-Real (FoR) dataset was used to train and test the proposed model. The Accuracy, F1-score, Precision, Recall, and Loss are used as metrics to evaluate the model's performance. The model performed well, with an accuracy of 99.64%. The proposed model was comprehensively evaluated by systematically varying the extracted features through different derivative orders, where the accuracy decreased to 94.34%. The experimental results demonstrate the effectiveness of the model in accurately detecting fabricated audio.

1. Introduction

The proliferation of social media platforms aids the dispersal of news and shared information more quickly, bringing many benefits by tying people across the globe. Unfortunately, these platforms have also been used as a weapon of misinformation by fake profiles, causing general unrest [1]. Fake audio is one of the technologies used to spread misinformation [2]. Audio manipulation technology was developed to improve and facilitate a humane existence, but it also opens up a wide range of negative uses, such as spreading fake information. The risk of utilizing altered audio to propagate misinformation worldwide is enhanced by the abundance of tools and programs available for altering voice, which can be accessed via a Personal Computer (PC) or a mobile device. Synthetic, replayed, and imitation-based are types of audio fakes [3]. This problem motivated us to research designing a model capable of accurately distinguishing between fake and real audio. Many techniques are used to detect fake audio, and often, they involve leveraging the advancements in artificial intelligence used to create them. Deep learning technology can effectively create fake audio with Text-To-Speech (TTS) and Voice Conversion (VC) techniques, making it challenging to discern between real and fake audio [4]. Recently, deepfake was used to generate fake audio perfectly, creating a challenge to determine whether the audio is fake or not. Deepfakes are a tool in computer science that uses Artificial Intelligence (AI) to generate or manipulate an image, audio, video, or text [3]. The primary difference between deepfakes and manual editing is that deepfakes are generated by AI and closely mimic real-world artefacts. The Mel Frequency Cepstral Coefficients (MFCC) scale offers a crucial advantage in detecting fake audio by providing a representation that aligns with human auditory perception, making it highly sensitive to subtle manipulations [5]. This sensitivity is combined with the compact and informative feature representation that MFCCs provide by translating audio into a Mel spectrogram. MFCCs can show Visual inconsistencies, such as unnatural transitions or spectral distortions, which become apparent, aiding in the identification of manipulated audio. Machine learning or deep learning can leverage the advantages extracted by MFCCs and empower them to accurately distinguish between real and fake audio, enhancing the robustness of deepfake detection systems. Researchers have presented various solutions to the problem of fake audio using various techniques. Reimao and Tzerpos proposed models to detect fake audio, where the models were Naive Bayes (NB), Support Vector Machine (SVM), Decision Tree (DT), and Random Forests (RF) [6]. They had the best accuracy using the SVM model, which was 73.46%. MFCC with machine learning was used by Vimal et al. to acknowledge emotions in speech and classify them into eight emotions, which were furious, unbiased, cool, ecstatic, astonished, fearful, shocked, and poignant [7]. They achieved the highest accuracy of 88.54% using the RF. Khochare et al. produced a comparison between two methods to distinguish between real and fake audio [8]. These methods were a feature-based approach and an image-based approach. The first method converted the audio in the dataset into various spectral features and fed them to a machine learning model to classify the audio as real or fake audio. The second method used the Mel spectrogram to get features and fed them to deep learning to classify the audio as real or fake audio. The best model in their work was the Temporal Convolutional Network (TCN). The paper by Hamza et al. used machine and deep learning-based approaches to detect fake audio, where the MFCC technique was used to acquire the most useful feature from the audio [9].

*Corresponding Author.
E-mail address: mohammed.m.arjeeli@uotechnology.edu.iq ; Tel: (+964) 772-253 8549 (Mohammed Msallam)

https://doi.org/10.30772/qjes.2025.161951.1608

Nomenclature

CNN	Convolutional Neural Network	EER	Equal Error Rate
MFCC	Mel Frequency Cepstral Coefficients	CFCCIF	Cochlear Filter Cepstral Coefficients and Instantaneous Frequency
FoR	Fake-or-Real	SV	Speaker Verification
PC	Personal Computer	RNN	Recurrent Neural Network
TTS	Text-To-Speech	APTLY	Processing Techniques Lab at York
VC	Voice Conversion	WAV	Waveform Audio File
AI	Artificial Intelligence	MPEG	Moving Picture Experts Group
NB	Naive Bayes	MP3	MPEG Audio Layer 3
SVM	Support Vector Machine	ASR	Automatic Speech Recognition
DT	Decision Tree	DFT	Discrete Fourier Transform
RF	Random Forests	DCT	Discrete Cosine transform
TCN	Temporal Convolutional Network	HW	Hamming Window

Spoofing attacks detected by Rahul et al., who used deep-convolutional neural networks [10]. They extracted features from speech using the MFCC. Equal Error Rate (EER) in their work was 0.9056%. In [11], the authors presented a countermeasure for those prone to attacks that originate from spoofed audio. They used an approach that extracted features by spectrograms and MFCC and built a modular architecture based on deep neural networks. The combination of Cochlear Filter Cepstral Coefficients and Instantaneous Frequency (CFCCIF) with MFCC was proposed by Patel and Patil to address threats to current Speaker Verification (SV) systems [12]. Ballesteros et al. proposed a model based on deep learning to classify fake and original voices [13]. They obtained histograms from recordings and used them as input for the Convolutional Neural Network (CNN). Their proposed system successfully detected fake voice content, as the accuracy of their work was 98.50%. Liu et al. used the Haas effect technique to create a stereo faking corpus [14]. They proposed two models to identify fake audio which were the MFCC and SVM model, and the CNN model. Rezaul et al. proposed a method to enhance audio classification using CNN and Recurrent Neural Network (RNN) Models, and they used MFCC to extract features [15]. Table 1 presents a summary of the best related work similar to our work, along with the value and Metrics used in those works.

Table 1. Summary of related works.

Ref.	Year	Best model	Metrics	Score
[6]	2019	SVM	Acc	73.46%
[7]	2021	RF	Acc	88.54%
[8]	2021	TCN	Acc	92.00%
[9]	2022	VGG16	Acc	93.00%
[10]	2020	ResNet34	EER	00.90%
[11]	2023	Neural Networks Assembly	EER	06.66%
[12]	2015	GMM	EER	01.52%
[13]	2021	CNN	Acc	98.50%
[14]	2021	CNN	Acc	99.00%
[15]	2024	RNN	Acc	96.52 %

The best model was RNN, with an accuracy rate of 96.52%. RNNs are adept at processing sequential data, but their application in fake audio detection is hampered by several issues. The vanishing and exploding gradient problem are the primary issue in RNNs. This issue limits their ability to capture and remember long-range temporal dependencies crucial for identifying subtle, dispersed manipulations in spoofed audio. This inherent sequential processing also makes RNNs difficult to parallelize, leading to significantly slower training times compared to architectures like CNNs, and they can struggle with limited memory when dealing with exceptionally long audio sequences, potentially overlooking crucial context. The research gap in identifying fake audio is that researchers presented models to predict fake audio without testing their models on data containing noise. Therefore, this paper contributes by presenting a new architecture for CNN that is used to detect fake audio with high efficiency to alleviate the concerns of fake audio posted on social media for the famous. It also presents a novel approach to evaluate the new CNN architecture by features overlap and training the CNN model on the overlapped features. The derivative is used to overlap features. So, the contribution of this article is as follows:

- A new architecture is proposed to detect fake audio to alleviate the concerns about misinformation.
- MFCC is used to extract features from audio.
- The complex mix is applied to the extracted features using a derivative to evaluate the proposed architecture of CNN.
- The effect of the derivative order on the CNN is studied.
- The proposed architecture succeeded in predicting the fake audio when features were mixed.

Following this introduction, Section 2 meticulously details the methodology employed in this research. This section provides a comprehensive description of the dataset utilized and preprocessing, along with a concise yet thorough explanation of the MFCC and CNN techniques applied in this work. Section 3 is dedicated to presenting the results obtained and engaging in a detailed discussion of their implications. Finally, Section 4 encapsulates the overarching conclusions drawn from this study.

2. Methodology

Detecting fake audio can be accomplished using a variety of machine learning approaches. Our proposed model for predicting fake audio is based on CNN, which is a type of deep learning that uses artificial neural networks to learn complex patterns and relationships in data. It is capable of processing large amounts of data and identifying complex relationships between features. Using CNN, the proposed model has layers with more neurons at the top and fewer neurons at the bottom. With the MFCC, the important features extracted from audio files that divided into 50% as the training set, 25% validation set, and 25% testing set. The proposed model is trained on the training set and tested on the validation set in the training phase. Following that, the testing phase is to test the CNN model on unseen test data to evaluate the performance of the model and compare it. Figure 1 shows a diagram of the overall processes to predict fake audio. In our proposal, the derivative is used to mix extracted features, and then the CNN model is trained and assessed on these mixed features to evaluate the proposed CNN model architecture.

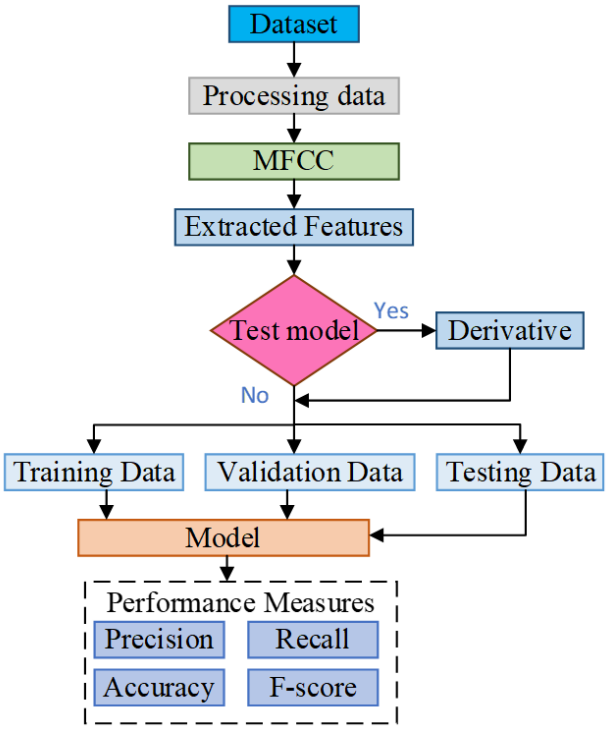


Figure 1. Overall processes to predict fake audio.

2.1 Dataset

The dataset, which is used in this work, was created by members of the Processing Techniques Lab at York (APTLY), which is available at https://bil.eecs.yorku.ca/datasets. There are four versions of the Fake-or-Real (FoR) dataset available for download, which are FoR-original, FoR-norm, FoR-rerec, and FoR-2sec. This work uses the first version of the FoR dataset, which is called for-original. It contains the audio files collected from the speech sources and has been without any modification. The for-original dataset has 69,321 samples of audio files with two classes. The first class is real, comprising 34,605 audio files. The second class is fake, comprising 34,716 audio files. Therefore, the data is balanced.

2.2 Processing

In preprocessing, there are two tasks, depending on the format and size of audio files, as shown in Fig. 2. The audio files with a 0-byte size were removed from the FoR dataset because they do not affect CNN learning. The audio files in the FoR dataset are in Waveform Audio File (WAV) and Moving Picture Experts Group (MPEG) Audio Layer 3 (MP3) format. The MP3 files were converted into WAV format. After processing the data stage, the FoR dataset has 34020 audio files as fake and 34605 audio files as real, so it is almost balanced. Figure 3 shows the distribution of fake and real FoR datasets.

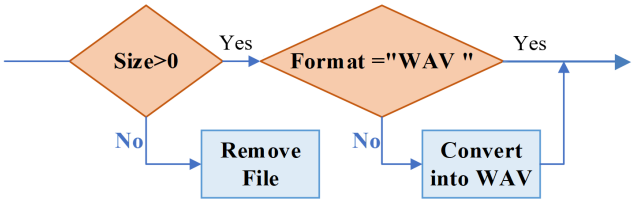


Figure 2. The preprocessing.

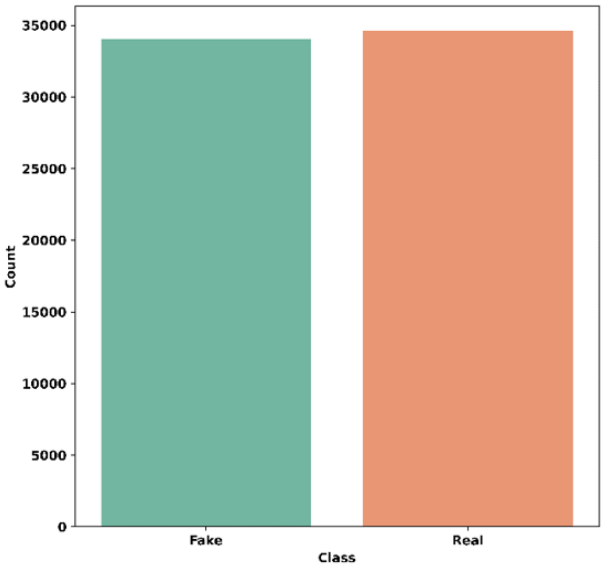
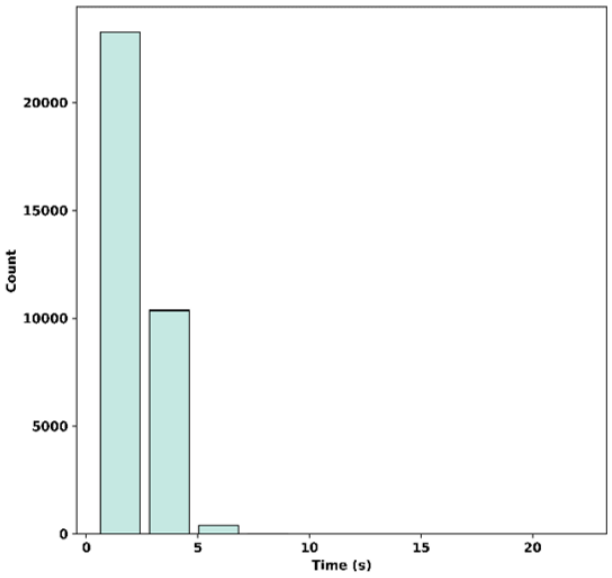
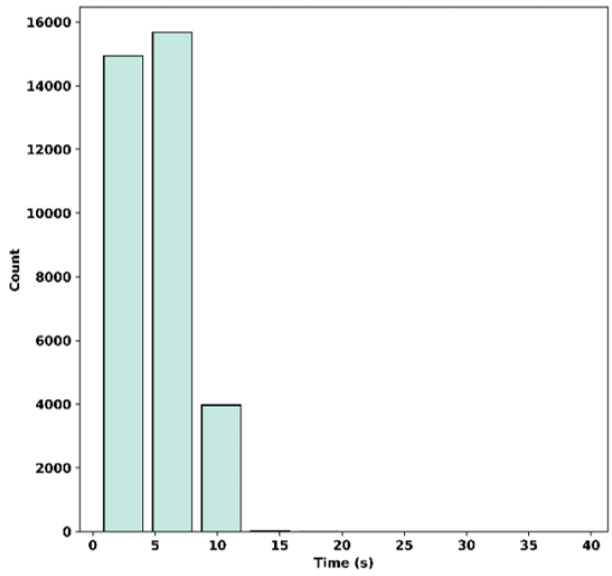


Figure 3. The distribution of fake and real.

The length of the audio files of the FoR dataset ranges between 0.4 sec and 39.8 sec. Most audio files of the FoR dataset are between 0.4 sec and 4 sec in fake audio files, as shown in Fig. 4a. Most audio files of the FoR dataset are between 0.4 sec and 10 sec in real audio files, as shown in Fig. 4b. The key statistical measures for the duration characteristics of real and fake audio files in the FoR dataset are visualized in Table 2, providing a clear picture of the duration of audio files in the FoR dataset.



(a) Model 01-Fake



(b) Model 02-Real

Figure 4. The distribution of audio duration.

Table 2. Characteristics of duration audio files.

Characteristics	Real	Fake
Mean	5.2190 sec	2.3521 sec
Median	4.8533 sec	2.2454 sec
Standard Deviation	2.2083 sec	0.8344 sec
Count	34605	34020
Minimum	0.4901 sec	0.4179 sec
Maximum	39.8080 sec	22.4653 sec

2.3 Mel frequency cepstral coefficient

Mel Frequency Cepstral Coefficients (MFCCs) are a crucial feature in audio processing, particularly in speech recognition. MFCCs are widely used in Automatic Speech Recognition (ASR) systems [16]. MFCCs provide a condensed and efficient means of expressing speech features, which facilitates computer comprehension of spoken language. MFCCs can also be used for speaker identification, as individual vocal tracts produce unique patterns of MFCC. The MFCC can be calculated by conducting six processes, which include pre-emphasis, framing and windowing, Discrete Fourier Transform (DFT), Mel Frequency filter bank, Logarithm, and Discrete Cosine transform

(DCT) as shown in Fig. 5. In the field of signal processing, pre-emphasis is a popular pre-processing technique used to compensate for the high frequency of the signal that was suppressed during signal production. The speech signal $x(k)$ can be sent through a high-pass filter. The high-pass filter is defined in Eq. 1, where $y(k)$ is the output signal and the value of a is usually between -0.97 and 1.0 [17].

$$y(k) = x(k) - ax(k-1) \tag{1}$$

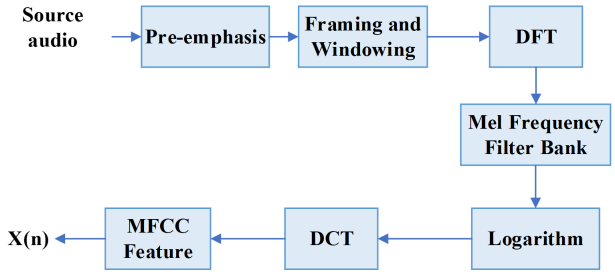


Figure 5. The MFCC steps.

The speech signal is separated into frames that are between 20 and 30 milliseconds in length and are adjacent to one another in framing processing. Windowing is a signal processing technique that divides an audio stream into temporal segments. The Hamming Window (HW) function is the window function that is typically utilized in MFCC. It keeps the opening and closing of the frame points consistent. The HW function is defined the Eq. 2, where $w(k)$ is the window function [17].

$$y(k) = x(k) \times w(k) \tag{2}$$

DFT is a mathematical tool that converts a discrete-time signal, such as a digital audio waveform, from its original time domain into a frequency domain representation where it is used to compute the power spectrum in an audio signal. DFT is a signal analysis method that allows for the extraction and compression of certain aspects of the audio signal without sacrificing any relevant information. The DFT is defined the Eq. 3, where x(k) is the discrete signal and K is the signal length [18]. Mel Frequency filter bank is a Mel band-pass filter that is constructed depending on pitch perception. The Mel filter was first created for speech analysis, and it aims to extract non-linear representations of the speech signal, much like the human ear does. The Mel Frequency filter bank is defined the Eq. **??**, where f (m) is the center frequency of the triangular filter [18].

$$x(n) = \sum_{k=0}^{K-1} x(k) \times e^{\frac{2\pi jnk}{K}}, \quad n = 1,2,3,...,K-1 \tag{3}$$

$$f(x) = \begin{cases} 0 \Rightarrow & \Rightarrow k < f(m-1) \\ \frac{k-f(m-1)}{f(m)-f(m-1)} \Rightarrow & \Rightarrow f(m+1) \leq k8 < f(m) \\ 1 \Rightarrow & \Rightarrow k = f(m) \\ \frac{f(m-1)-k}{f(m+1)-f(m)} \Rightarrow & \Rightarrow f(m) <\leq< f(m+1) \\ 0 \Rightarrow & \Rightarrow k > f(m+1) \end{cases} \tag{4}$$

The logarithm plays a crucial role within the MFCC computation process, primarily because of its relationship to how humans perceive sound. The logarithm is used to convert normal frequency into Mel frequency because it is a psychoacoustic scale that was designed to replicate the non-linear human. The center frequency of the filter bank is computed by approximating the Mel scale, which is given by Eq. 5, where the normal frequency is f and Mel frequency is m [19].

$$m = 2595\ log_{10}\left(\frac{f}{700} + 1\right) \tag{5}$$

The DCT considers a finite sequence of points about a summation of cosine functions at various frequencies. The key advantages of the DCT are its energy compaction property, to decorrelate the outputs of the Mel filter bank, and to perform dimensionality reduction. Nasir Ahmed introduced DCT in 1972 [18]. The DCT is applied to the Mel filter bank in the MFCC process to separate the logarithmic spectral magnitude relationship from the filter bank or to choose the most accelerative coefficients. The DCT is computed by Eq. 6, where $x(k)$ is the discrete signal and K is the signal length [18].

$$x(n) = \sum_{k=0}^{K-1} x(k)\ cos\left(\frac{2\pi jnk}{K}\right), \quad n = 1,2,3,...,K-1 \tag{6}$$

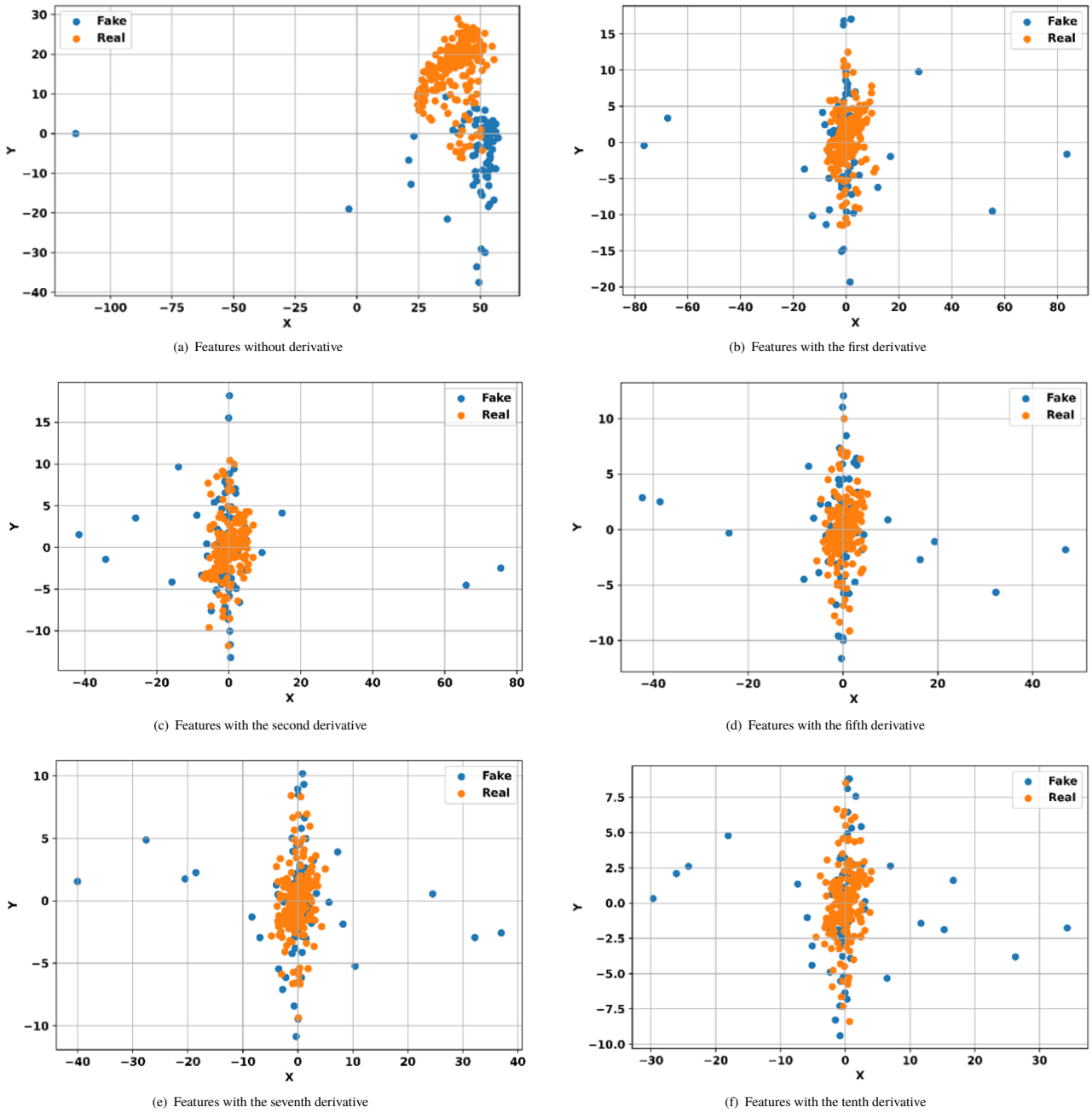
The derivative order is applied to MFCC features to mix and overlap the extracted features using MFCC. This step is used to evaluate the proposed architecture of the CNN and to determine the performance of the proposed architecture of the CNN when overlapped features are used. The function of CNN is to find a relationship between features to classify data based on them. The derivative order increases, the data becomes more confused, and it becomes more difficult for the CNN to find this relationship. The derivative of MFCC features is computed by Eq. 7, where n is the number of times the derivative is taken.

$$\frac{d^m x(t)}{dt} = \frac{d^m}{dt} x(t) \tag{7}$$

Fig. 6 shows the effect of the derivative order on the distribution features of two audio files, where one is fake and the other is real. The number of cepstrum features to return was two, where the first represents the x-axis and the second represents the y-axis. The distributed features without derivatives are separate, as shown in Fig. 6a. However, when the derivative order is increased, the features overlap more, as shown in Fig. 6d.

2.4 Convolutional neural network

A Convolutional Neural Network (CNN), which is considered one type of machine learning technique that includes many machine learning techniques, such as SVM [20]. CNN utilizes artificial neural networks to create a machine learning model [21]. CNN was proposed by Hubel and Wiesel in 1962 [22]. The function and structure of the human brain serve as inspiration for artificial neural networks, which can recognize complex patterns from large amounts of data [23]. The CNN model is made up of numerous layers, where the order of these layers is called the CNN model architecture [24]. The layer in a CNN consists of nodes, which are connected, resulting in a dense web of connections that promotes system efficiency and redundancy [25]. Each CNN layer extracts a different degree of abstraction from the input [26]. A CNN model is trained based on huge sets of labelled data that are provided for each class of the model. As a result of this training, the model understands how the input data relates to the truth state. The trained model can forecast the outcome of unobserved input. The size of extracted features using MFCC is 400 x 10, and it is resized to become 400 x 10 x 1 for each sample of the audio files. Thereby, the first layer of the proposed architecture is an input layer (InputLayer) with size 400 x 10 x 1. The architecture of a CNN designed to detect fake audio is a sequential arrangement of layers, as shown in Fig. 7. The next layers used in this work, after the input layer, are a Convolutional 2-Dimensional (Conv2D), Max Pooling 2-Dimensional (MaxPooling2D), Flatten, and Dense layers, each performing a distinct function. The Conv2D layer is the foundational building block in this work, where learnable filters are convolved with the input data to automatically extract spatial hierarchies of features, producing a set of feature maps [21]. The MaxPooling2D layer is used to downs ample the feature maps by selecting the maximum value within a defined region. The reason for using in the proposed model the MaxPooling2D layer is to decrease the number of parameters and computational cost in the proposed model, while also providing a degree of representation invariance, making the network more robust to minor shifts in the input [25]. The Flatten layer resizes the multi-dimensional output from the MaxPooling2D layers into a one-dimensional vector to bridge the gap between the feature extraction part of the previous layers and the final classification part. Finally, the Dense layer receives this flattened vector and uses it to perform the final classification task. The Conv2D and Dense layers need to define an activation function, where Table 3 shows the details of the layers in the proposed architecture of the CNN.



(a) Features without derivative

(b) Features with the first derivative

(c) Features with the second derivative

(d) Features with the fifth derivative

(e) Features with the seventh derivative

(f) Features with the tenth derivative

Figure 6. Effect of derivatives on distributing features.

Table 3. Summary of related works.

Name layer	Type layer	Activation function	Input size	Output size
Input	InputLayer	—	$400 \times 10 \times 1$	$400 \times 10 \times 1$
$Hidden_1$	Conv2D	ReLU	$400 \times 10 \times 1$	$398 \times 8 \times 32$
$Hidden_2$	MaxPooling2D	—	$398 \times 8 \times 32$	$199 \times 4 \times 32$
$Hidden_3$	Conv2D	ReLU	$199 \times 4 \times 32$	$197 \times 2 \times 64$
$Hidden_4$	MaxPooling2D	—	$197 \times 2 \times 64$	$98 \times 1 \times 64$
$Hidden_5$	Flatten	—	$98 \times 1 \times 64$	6272
$Hidden_6$	Dense	ReLU	6272	512
Output	Dense	Sigmoid	512	1

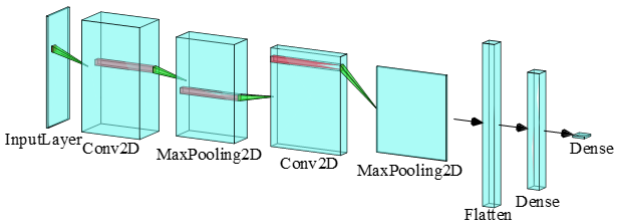


Figure 7. Proposed architecture of the CNN model.

3. Results and discussion

The dataset was divided into validation data, which was 25%, testing data, which was 25%, and training data, which was 50%. The proposed model of CNN is trained by using 50 epochs with 10 batch sizes. Smaller batch sizes are preferred because bigger batch sizes result in faster training, but overfitting,

whereas smaller batch sizes can help avoid overfitting but result in slower training [27]. The proposed model utilizes Adaptive Moment Estimation (ADAM) to adjust the learnable parameters. The loss function used in this work is binary cross-entropy, which is used in binary classification problems and measures the performance of a classification model whose output is a probability value between 0 and 1. It quantifies the difference between the predicted probability and the actual true label, as illustrated in Eq. 8 [28].

$$Loss = \frac{1}{N}\sum_{i=1}^{N}\left(-\left(y\ log(\hat{y}) + (1-y)\ log(1-\hat{y})\right)\right) \quad (8)$$

Where y is the true label in binary classification problems, $\hat{y}$is the predicted probability by the model, and N is the number of observations. Fig. 8 illustrates the training and validation loss curves for the proposed model over 50 epochs. The black line, representing the training loss, demonstrates a rapid and monotonic decrease, stabilizing near zero after approximately 20 epochs. This indicates that the model is successfully minimizing the error on the training dataset. Conversely, the blue line, representing the validation loss, initially decreases in parallel with the training loss but begins to ascend steadily after epoch 20. The widening divergence between the two curves is a standard representation of overfitting, where the proposed model learns on the training data too well, thereby compromising its ability to generalize effectively to new, unseen data.

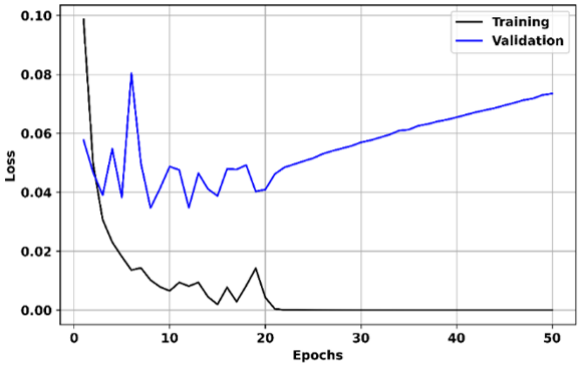


Figure 8**.** Training and validation loss.

Figure 9 shows the training and validation accuracy of a machine learning model over 50 epochs. The black line represents the training accuracy, which shows a steep increase in the initial epochs, quickly approaching 1.0, indicating that the model is effectively learning the patterns from the training data. The blue line represents the validation accuracy, which rises rapidly at first but then stabilizes at approximately 0.994 after around 20 epochs. A significant observation is the divergence between the two curves after epoch 20, where the training accuracy reaches a perfect 1.0 while the validation accuracy plateaus. This performance gap suggests that the model is beginning to overfit the training data, meaning it is memorizing the specific examples rather than generalizing well to unseen data.

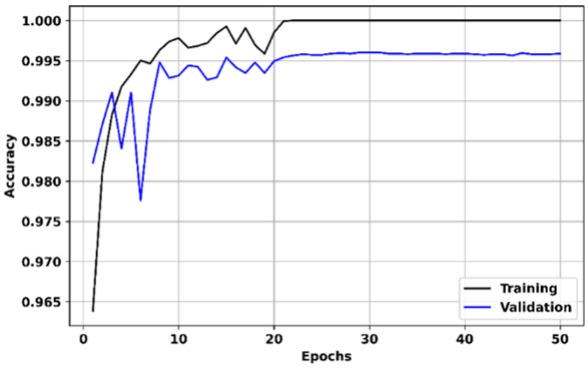


Figure 9**.** Training and validation accuracy.

The performance of our model was assessed using the confusion matrix, a square table used to assess a classification performance model. The test data were 17157 samples, of which 8449 were of the fake audio class and 8708 were of the real audio class. In our model, the prediction was 8668 for the real audio class, which is the correct prediction, while it predicted that 40 was the fake audio class, which is the incorrect prediction. It also predicted that 22 were the fake audio class, which is the incorrect prediction, while it predicted that 8427 were the real audio class, which is the correct prediction, as shown in Fig. 10. In the last layer of the model, the probability that the testing sample will belong to a class is the output. If the value is greater than or equal to 0.5, the class that is predicted is the fake audio class; if not, it is the real audio class. Most values of the fake audio class are close to one, which is more than 0.99, while other values are between 0.5 and 0.99. Most values of the fake real class are remarkably close to zero, which is less than 0.1, while other values are between 0.1 and 0.5. If the probability values are close to 0.5, it means that the prediction possesses a high degree of suspicion, and if the value of probability value is far from 0.5, it means that the prediction possesses a high degree of certainty. Figure 11 shows values of probability between 0.46 and 0.56, where the prediction possesses a high degree of suspicion. The proposed model had 99.64% accuracy on testing data in detecting fake audio, where the accuracy is the ratio between the count of correct predictions and the total count of predictions on testing data. The values of F1-score, precision, recall, and loss on test data are in Table 4.

Table 4**.** Results of the proposed model.

Metrics	*Value*
Accuracy	0.9964
F1-score	0.9963
Precision	0.9963
Recall	0.9974
Loss	0.0263

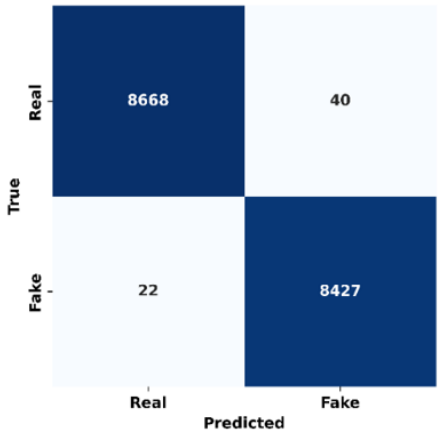


Figure 10**.** Confusion Matrix of predicted and truth classes.

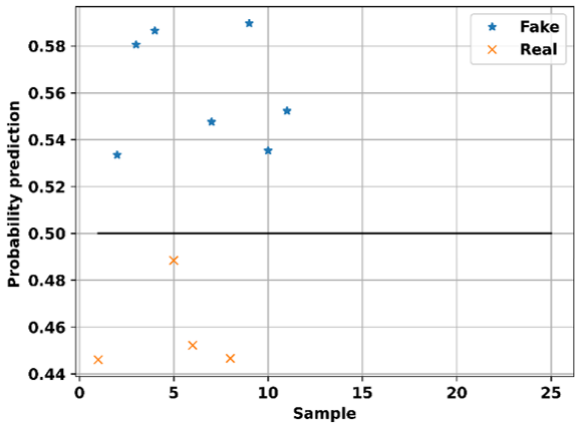


Figure 11**.** Probability values for the proposed model.

The extracted features using MFCC were derived by various derivative orders to overlap features. The derivative orders were from zero to twelve. Each derivative order created a new dataset. These datasets were used to train and

test the proposed model with the same parameters. Figure 12 illustrates the effect of the derivative order on accuracy, showing an inverse relationship between the derivative order and accuracy. The highest accuracy is achieved at order *do* (the original data), indicating that the original data contains the most useful, uncorrupted information. As the derivative order increases, accuracy steadily declines, due to noise amplification, overfitting, and loss of relevant information, suggesting that higher-order derivatives introduce more noise than useful information for this specific model and dataset. Figure 13 illustrates the relationship between the order of a derivative and loss.

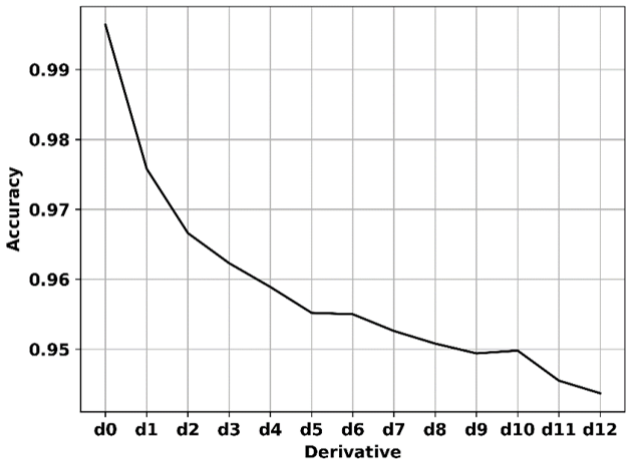


Figure 12**.** Differential effect of extracted features on the accuracy of the proposal model.

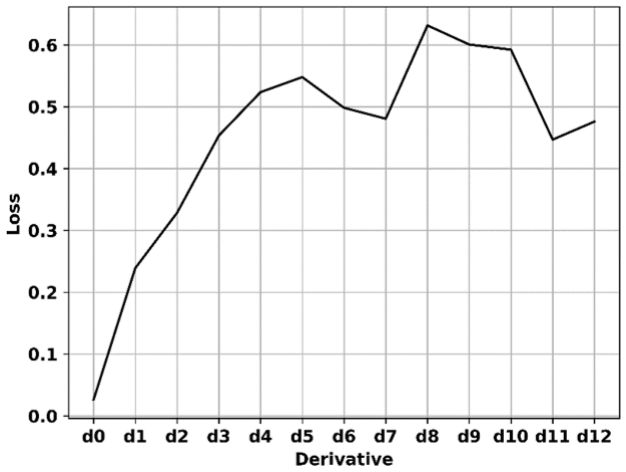


Figure 13**.** Differential effect of extracted features on the loss of the proposal model.

Additionally, the outcomes of the models that were acquired in the baseline for the same conditions are compared with the proposed model. The comparison results are listed in Table 5, where the proposed model outperformed other work in terms of prediction accuracy.

Table 5**.** The performance comparison.

Metrics	*Value*
[6]	76.78%
[8]	92.00%
[29]	94.21%
[30]	96.86%
This work	99.64%

4. Conclusions

In this work, a new architecture was proposed to detect fake audio to alleviate the concerns about the misinformation posted on social media. In this work, Features were extracted from audio using the MFCC. The derivative of extracted features is used to mix these features to evaluate the proposed architecture of CNN. The proposed architecture of CNN consisted of eight layers in sequential form. The performance of the proposed architecture of CNN was well, where it achieved an accuracy of 99.64%. Accuracy gradually decreases with increasing derivative order of MFCC features, because of noise amplification. It can be concluded that higher-order derivatives introduce more noise than useful representation into the proposed architecture of CNN. Future work can focus on hybrid CNN-RNN or transformer-based architectures to better capture temporal dependencies in audio files. Furthermore, a deeper investigation into feature engineering is warranted, exploring the integration of alternative or complementary audio features such as Chroma or Spectral Contrast to provide a richer input representation.

Authors' contribution

Mohammed M. Msallam conceived and designed the model of CNN and the experiments, performed the experiments, and performed the computation work. Esraa Y. Tarkan prepared figures and tables, authored and reviewed drafts of the article. Sarah K. Salim analyzed the data, ensuring the accuracy of the results, and approved the final draft.

Declaration of competing interest

The authors declare no conflicts of interest.

Funding source

This study didn't receive any specific funds.

Data availability

The data that support the findings of this study are available from the corresponding author upon reasonable request.

REFERENCES

[1] S. C. Venkatesh, S. Shaji, and B. M. Sundaram, "A fake profile detection model using multistage stacked ensemble classification," *Proceedings of Engineering and Technology Innovation*, vol. 26, pp. 18–32, 2024. [Online]. Available: https://doi.org/10.46604/peti.2024.13200

[2] E. Aïmeur, S. Amri, and G. Brassard, "Fake news, disinformation and misinformation in social media: A review," *Social Network Analysis and Mining*, vol. 13, p. 30, 2023. [Online]. Available: https://doi.org/10.1007/s13278-023-01028-5

[3] Z. Khanjani, G. Watson, and V. P. Janeja, "Audio deepfakes: A survey," *Frontiers in Big Data*, vol. 5, p. 1001063, 2023. [Online]. Available: https://doi.org/10.3389/fdata.2022.1001063

[4] A. Alali and G. Theodorakopoulos, "Review of existing methods for generating and detecting fake and partially fake audio," in *Proceedings of the 10th ACM International Workshop on Security and Privacy Analytics*, 2024. [Online]. Available: https://doi.org/10.1145/3643651.3659894

[5] A. Jellali, I. Ben Fredj, and K. Ouni, "Pushing the boundaries of deepfake audio detection with a hybrid mfcc and spectral contrast approach," *Multimedia Tools and Applications*, pp. 1–20, 2024. [Online]. Available: https://doi.org/10.1007/s11042-024-19819-z

[6] R. Reimao and V. Tzerpos, "FoR: A dataset for synthetic speech detection," in *10th International Conference on Speech Technology and Human-Computer Dialogue*, 2019. [Online]. Available: https://doi.org/10.1109/SPED.2019.8906599

[7] B. Vimal, M. Surya, Darshan, V. S. Sridhar, and A. Ashok, "MFCC based audio classification using machine learning," in *12th International Conference on Computing Communication and Networking Technologies*, 2021. [Online]. Available: https://doi.org/10.1007/s43926-023-00049-y

[8] J. Khochare, C. Joshi, B. Yenarkar, S. Suratkar, and F. Kazi, "A deep learning framework for audio deepfake detection," *Arabian Journal for Science and Engineering*, vol. 47, no. 3, pp. 3447–3458, 2021. [Online]. Available: https://doi.org/10.1007/s13369-021-06297-w

[9] A. Hamza, A. R. R. Javed, F. Iqbal, N. Kryvinska, A. S. Almadhor, and Z. Jalil, "Deepfake audio detection via mfcc features using machine learning," *IEEE Access*, vol. 10, pp. 134 018–134 028, 2022. [Online]. Available: https://doi.org/10.1109/ACCESS.2022.3231480

[10] T. P. Rahul, P. R. Aravind, C. Ranjith, U. Nechiyil, and N. Parampa-rambath, "Audio spoofing verification using deep convolutional neural networks by transfer learning," 2020, arXiv preprint arXiv:2008.03464. [Online]. Available: https://doi.org/10.48550/arXiv.2008.03464

[11] C. A. Hernández-Nava, E. A. Rincón-García, P. Lara-Velázquez, S. Gerardo de-los Cobos-Silva, M. A. Gutiérrez-Andrade, and R. A. Mora-Gutiérrez, "Voice spoofing detection using a neural networks assembly considering spectrograms and mel frequency cepstral coefficients," *PeerJ Computer Science*, vol. 9, p. e1740, 2023. [Online]. Available: https://doi.org/10.7717/peerj-cs.1740

[12] T. B. Patel and H. A. Patil, "Combining evidences from mel cepstral, cochlear filter cepstral and instantaneous frequency features for detection of natural vs. spoofed speech," in *Interspeech*, 2015, pp. 2062–2066.

[13] D. M. Ballesteros, Y. Rodriguez-Ortega, D. Renza, and G. Arce, "Deep4snet: Deep learning for fake speech classification," *Expert Systems with Applications*, vol. 184, p. 115465, 2021. [Online]. Available: https://doi.org/10.1016/j.eswa.2021.115465

[14] T. Liu, D. Yan, R. Wang, N. Yan, and G. Chen, "Identification of fake stereo audio using SVM and CNN," *Information*, vol. 12, no. 7, p. 263, 2021. [Online]. Available: https://doi.org/10.3390/info12070263

[15] K. M. Rezaul, M. Jewel, M. S. Islam, K. N. A. Siddiquee, N. Barua, M. A. Rahman, M. Shan-A-Khuda, R. B. Sulaiman, M. S. I. Shaikh, M. A. Hamim, F. M. Tanmoy, A. U. Haque, M. S. Nipun, N. Dorudian, A. Kareem, A. K. Farid, A. Mubarak, T. Jannat, and U. F. T. Asha, "Enhancing audio classification through mfcc feature extraction and data augmentation with cnn and rnn models," *International Journal of Advanced Computer Science and Applications*, vol. 15, no. 7, pp. 37–53, 2024. [Online]. Available: http://dx.doi.org/10.14569/IJACSA.2024.0150704

[16] M. Malik, M. K. Malik, K. Mehmood, and I. Makhdoom, "Automatic speech recognition: A survey," *Multimedia Tools and Applications*, vol. 80, no. 6, pp. 9411–9457, 2021. [Online]. Available: https://doi.org/10.1007/s11042-020-10073-7

[17] M. Altayeb and A. Arabiat, "Crack detection based on mel-frequency cepstral coefficients features using multiple classifiers," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 14, no. 3, pp. 3332–3341, 2024. [Online]. Available: https://doi.org/10.11591/ijece.v14i3.pp3332-3341

[18] Z. K. Abdul and A. K. Al-Talabani, "Mel frequency cepstral coefficient and its applications: A review," *IEEE Access*, vol. 10, pp. 122 136–122 158, 2022. [Online]. Available: https://doi.org/10.1109/ACCESS.2022.3223444

[19] S. Sigurdsson, K. B. Petersen, and T. Lehn-Schiøler, "Mel frequency cepstral coefficients: An evaluation of robustness of mp3 encoded music," in *7th International Conference on Music Information Retrieval*, 2006, pp. 286–289.

[20] S. A. Alrubaie and I. M. Hassoon, "Support vector machine (svm) for colorization the grayscale image," *Al-Qadisiyah Journal for Engineering Sciences*, vol. 13, no. 3, pp. 207–214, 2021. [Online]. Available: https://doi.org/10.30772/qjes.v13i3.658

[21] L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaria, M. A. Fadhel, M. Al-Amidie, and L. Farhan, "Review of deep learning: Concepts, cnn architectures, challenges, applications, future directions," *Journal of Big Data*, vol. 8, no. 1, 2021. [Online]. Available: https://doi.org/10.1186/s40537-021-00444-8

[22] N. A. Abdulrazzaq and A. M. Radhi, "Face recognition using deep convolutional neural networks," *Al-Qadisiyah Journal for Engineering Sciences*, vol. 18, no. 3, 2025.

[23] S. Sharma, S. Sharma, and A. Athaiya, "Activation functions in neural networks," *International Journal of Engineering Applied Sciences and Technology*, vol. 04, no. 12, pp. 310–316, 2020. [Online]. Available: https://www.geeksforgeeks.org/

[24] S. K. Salim, M. M. Msallam, and H. Ismail, "Design novel cnn architecture to protect personal devices from unauthorized access using face recognition," *Nanotechnology Perceptions*, vol. 20, no. 3, pp. 82–93, 2024. [Online]. Available: https://doi.org/10.62441/nano-ntp.v20i3.7

[25] O. A. Olayemi, O. I. Salako, A. Jinadu, A. M. Obalalu, and B. E. Anyaegbuna, "Aerodynamic lift coefficient prediction of supercritical airfoils at transonic flow regime using convolutional neural networks (cnns) and multi-layer perceptions (mlps)," *Al-Qadisiyah Journal for Engineering Sciences*, vol. 16, no. 2, pp. 108–115, 2023. [Online]. Available: https://kwasuspace.kwasu.edu.ng/handle/123456789/272

[26] A. M. Abood, A. R. Nasser, and H. Al-Khazraji, "Predictive maintenance of electromechanical systems based on enhanced generative adversarial neural network with convolutional neural network," *International Journal of Artificial Intelligence*, vol. 12, no. 4, pp. 1704–1712, 2023. [Online]. Available: https://doi.org/10.11591/ijai.v12.i4.pp1704-1712

[27] Z. Li, E. Wallace, S. Shen, K. Lin, K. Keutzer, D. Klein, and J. E. Gonzalez, "Train large, then compress: Rethinking model size for efficient training and inference of transformers," in *37th International Conference on Machine Learning*, 2020, pp. 5914–5924. [Online]. Available: https://doi.org/10.48550/arXiv.2002.11794

[28] Y. Ho and S. Wookey, "The real-world-weight cross-entropy loss function: Modeling the costs of mislabeling," *IEEE Access*, vol. 8, pp. 4806–4813, 2020.

[29] C. Ahmadi, S. H. Wang, S. P. Chiu, and J. L. Chen, "Dual acoustic feature fusion for enhanced audio deepfake detection using vgg-16 architecture: Mitigating speech tampering with mfcc and eltp," in *2024 RIVF International Conference on Computing and Communication Technologies (RIVF)*, 2024, pp. 216–220.

[30] N. Bakken, S. Singh, M. Prashant, and T. Das, "Deep fake audio detection framework using mfccs, chroma features, and spectrogram images," in *2025 IEEE Conference on Artificial Intelligence (CAI)*, 2025, pp. 1–6. [Online]. Available: https://doi.org/10.1109/CAI64502.2025.00175

How to cite this article:
Esraa Y. Tarkan, Mohammed M. Msallam, and Sarah K. Salim, (2026). 'Detecting fake audio using convolutional neural networks for reducing misinformation', Al-Qadisiyah Journal for Engineering Sciences, 19(2), pp. 270- 277. https://doi.org/10.30772/qjes.2025.161951.1608