Signal, Image and Video Processing (2024) 18:2117–2133
https://doi.org/10.1007/s11760-023-02856-w

ORIGINAL PAPER

Detecting audio copy-move forgery with an artificial neural network

Fulya Akdeniz[1] · Yaşar Becerikli[1]

Received: 23 September 2022 / Revised: 11 October 2023 / Accepted: 17 October 2023 / Published online: 11 January 2024

Abstract

Given how easily audio data can be obtained, audio recordings are subject to both malicious and unmalicious tampering and manipulation that can compromise the integrity and reliability of audio data. Because audio recordings can be used in many strategic areas, detecting such tampering and manipulation of audio data is critical. Although the literature demonstrates the lack of any accurate, integrated system for detecting copy-move forgery, the field shows great promise for research. Thus, our proposed method seeks to support the detection of the passive technique of audio copy-move forgery. For our study, forgery audio data were obtained from the TIMIT dataset, and 4378 audio recordings were used: 2189 of original audio and 2189 of audio created by copy-move forgery. After the voiced and unvoiced regions in the audio signal were determined by the yet another algorithm for pitch tracking, the features were obtained from the signals using Mel frequency cepstrum coefficients (MFCCs), delta (Δ) MFCCs, and $\Delta\Delta$MFCCs coefficients together, along with linear prediction coefficients (LPCs). In turn, those features were classified using artificial neural networks. Our experimental results demonstrate that the best results were 75.34% detection with the MFCC method, 73.97% detection with the ΔMFCC method, 72.37% detection with the $\Delta\Delta$MFCC method, 76.48% detection with the MFCC + ΔMFCC + $\Delta\Delta$MFCC method, and 74.77% detection with the LPC method. Using the MFCC + ΔMFCC + $\Delta\Delta$MFCC method, in which the features are used together, we determined that the models give far superior results even with relatively few epochs. The proposed method is also more robust than other methods in the literature because it does not use threshold values.

Keywords Digital multimedia forensics · Audio forensics · Audio tampering · Audio copy-move forgery · Artificial neural network · Digital multimedia security

1 Introduction

Digital multimedia, including audio, image, animation, and video elements, represent one of the most important areas of research today precisely because smart devices and internet resources required to access those elements are now cheap and easily available. However, such ease of accessibility has advantages as well as disadvantages for users. Although people without professional training in using audio, image, and video media can easily make changes to audio, image, and video files by using smartphones, other smart devices, and various web applications, all changes made to those files, or recordings, break the integrity, authenticity, and security of the data therein [1–3]. Because such changes threaten biometric data (e.g., fingerprints, irises, faces, and voices) required for users to access digitally secure applications, different methods have been proposed to ensure the security of digital multimedia data [4].

In recent years, voice signals have been the most preferred of all forms of biometric data due to their outstanding performance and easy accessibility [5]. Especially in context involving human health (e.g., hygiene and infection), many industries have even accelerated the use of voice data for authentication [6]. However, modifications made to digital audio can easily alter the meaning of the speech or voice messages. Such modifications are made in various situations, including in response to terrorist incidents, in criminal cases, for blackmail, in music industry copyrights, in banking, for fake evidence in court, in cyber-physical systems, and in the deception of automatic speaker verification systems. Audio forgery is common precisely because audio recordings are easily obtained and quickly manipulated [7]. For that reason,

✉ Fulya Akdeniz
fulya.akdeniz@kocaeli.edu.tr

Yaşar Becerikli
ybecerikli@kocaeli.edu.tr

[1] Department of Computer Engineering, Kocaeli University, Kocaeli, Turkey

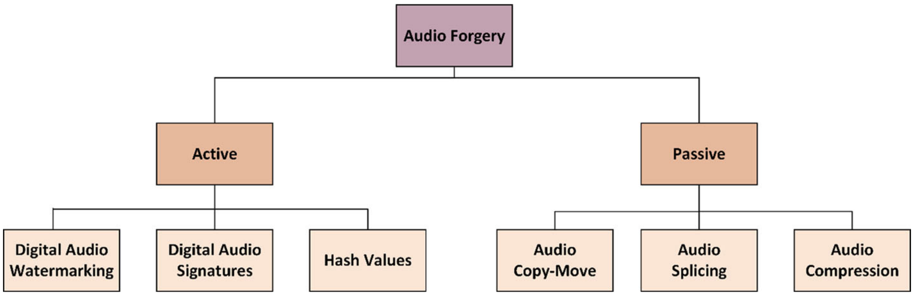


Fig. 1 Types of audio forgery techniques

verifying the integrity of audio recordings, as well as speaker verification, is an essential topic in research [1, 2, 8].

Audio forensics is a branch of science that includes archiving, analyzing, and examining audio recordings for presentation as evidence in court or official institutions. Therein, digital audio forensics detection is used to confirm authentication [9, 10] and has become increasingly important in forensics, especially digital multimedia forensics, in security- and military-intensive fields. Digital audio forensics is divided into two branches: active forensics and passive forensics. In active forensics, specific information (e.g., watermarks, hash values, and digital signatures) is placed in the original audio using various techniques in three categories: digital audio watermarking, digital audio signatures, and hash values. In the active approach, the integrity of audio signals is controlled in order to detect whether any forgery has occurred to the audio recordings by examining whether the actively embedded information (e.g., a watermark) has been removed. By contrast, passive forensics focuses on audio signals and their properties, and the audio data do not contain any extra special information. Passive techniques can also be divided into three categories: audio copy-move, audio splicing, and audio compression.

Although audio watermarks and signatures are used in authentication processes, they will be insufficient in many real-life circumstances because those operations require additional information. In real life, digital watermarks and signatures cannot be found in many audio signals, and it is unrealistic to expect such embedded information to be involved in all audio signals. Therefore, research on passive forensics, including digital audio blind forensics, is critical in the broader field of audio forensics [11–15], not least because passive techniques do not contain any additionally embedded information and are thus are more difficult to detect than active ones [8]. In our study, we aimed to detect one of those passive techniques, namely audio copy-move forgery (Fig. 1).

In the passive technique of audio copy-move forgery, some parts of an audio recording are copied and moved to different locations in the same recording [16]. Because the region copied from the recording is derived from the same recording, the forgery is highly difficult to detect, even by experts [17]. As a consequence of the technique, the integrity and reliability of the original audio recording are compromised, and the meaning of the speech in the recording is easily changed. Even though detecting audio copy-move forgery is critical, it has not been sufficiently addressed in the literature, and many researchers have stressed that despite studies on digital image forensics and video image forensics, studies on audio forensics remain in their infancy [11, 18].

In previous years, audio authentication processes involved examining irregularities and changes in frequency and time, such as captured in spectrograms. However, with the development of software and applications that facilitate professional audio forgery, audio forgery has become nearly impossible to detect audibly or visually. Moreover, of all audio forgery methods, copy-move forgery is one of the most dangerous [8]. Nevertheless, as many researchers have emphasized, a system for detecting audio copy-move forgery does not exist despite being urgently needed, and the number of studies on the topic remains disappointingly low [7, 8, 11, 16, 17, 19–21].

Considering all of the above, the chief contributions of our paper are as follows:

- This paper is the first to use Mel frequency cepstrum coefficients (MFCCs), delta (Δ) MFCCs, $\Delta\Delta$MFCCs, MFCC + ΔMFCC + $\Delta\Delta$MFCCs, and linear prediction coefficients (LPCs) with an artificial neural network (ANN) model and to comprehensively analyze the results for different hidden layers, different neurons therein, different batch sizes, and different numbers of epochs and present the results in accuracy and loss graphs.
- All models developed for the paper use audio data directly without applying any signal preprocessing to detect audio copy-move forgery.
- Unlike in nearly all studies on copy-move forgery, in which a threshold value was determined that could identify copy-move forgery in audio signals, audio copy-move forgery was detected in our work without using any threshold value and thus by depending less on the person, gender, audio recorder, and environmental conditions.

- Our experimental results show that the proposed methods succeeding in detecting the audio copy-move forgery without using any threshold value.

Based on our study, we propose a model to detect copy-move forgery in audio signals. The model, using MFCCs, LPCs, and an ANN, can robustly detect copy-move forgery without determining any threshold value. That property provides flexibility in reducing the dependence on the person, gender, voice recorder, and environmental conditions.

In the sections that follow, we explain related work in Sect. 2. After describing our study's materials and methods in Sect. 3, we detail the database, experimental setup, and experimental results in Sect. 4. We discuss the proposed model in Sect. 5 before offering our conclusions in Sect. 6.

2 Related work

As shown in the literature, research on audio forgery has continuously increased, especially on the topics of classifying audio recorders and microphones, identifying and recognizing speakers, and detecting audio splicing, synthesized speech, and/or audio compression. Nevertheless, studies on detecting audio copy-move forgery have been few.

In one study, Akdeniz et al. detected copy-move forgery in audio signals divided into voiced and unvoiced segments from which they extracted MFCC-based features. Using 1800 pieces of data obtained from the TIMIT dataset, they employed the Pearson correlation coefficient (PCC) to calculate the similarity between the features. By comparing the similarity between those coefficients according to a certain threshold value, they detected audio copy-move forgery [22]. In a similar study using 1400 pieces of data from the TIMIT dataset, Akdeniz et al. detected copy-move forgery in audio signals also divided into voiced and unvoiced segments. They obtained LPC features from those segments and used PCCs to calculate the similarity between the features according to a determined threshold value and thus detected audio copy-move forgery [23].

Meanwhile, Su et al. divided 500 tampered and 500 original audio files from the Chinese speech and LibriSpeech datasets into syllables using a pitch tracking algorithm as a means to detect copy-move forgery. They extracted features using constant Q cepstral coefficients and sliding window methods. In that process, they designated a threshold value according to the error value and used PCCs to calculate the similarity between the features according to a determined threshold value and thus detected audio copy-move forgery. In their study, they used a different attack: no attack on the signal, compressing, resampling, adding noise, changing format, and low-pass filtering [24]. Beyond that, Li et al. have proposed a method based on the pitch feature of audio signals to detect copy-move forgery. In their method, using 500 attack sound recordings, they first preprocessed the audio files nonlinearly and subsequently extracted the pitch sequence as a feature of the audio signals. Then, by comparing the difference between the syllables with the threshold value, they resolved whether there was a copy-move forgery in the signals. In their study, they used a different attack: no attack on the signal, compressing amplitude, resampling, adding Gaussian white noise, and changing format [20].

In other works, Yan et al. used the wall street journal speech database and the TIMIT dataset to generate copy-move forgery data. After dividing audio recordings into voiced and unvoiced audio segments, they extracted features of the pitch sequence and the LPC-based format sequence from the audio regions and calculated the similarity of the feature sets using the dynamic time warping method. By comparing the similarity values that they calculated with a threshold value, they detected copy-move forgery in the audio recordings [16]. By contrast, using 1000 original copy-move audio data generated in their research, Xie et al. obtained gammatone features, MFCC features, pitch features, and discrete Fourier transform (DFT) coefficients from the audio signals. They next used the C4.5 decision tree and neural network method to detect audio copy-move forgery, along with PCCs and the average difference method to calculate the similarity between the audio segments. Their study however, was the extra processing step needed to convert the signal data to spectrum images [21]. Imran et al. first extracted words using a voice activity detection (VAD) module to perform anterior segmentation. Employing forgery audio data that they created using King Saud University's Arabic Speech Database and generating 1350 forgery data, they obtained 1D-LBP features from the signals to detect copy-move forgeries and their locations. Ultimately, with that method, histograms of those words were created [8]. In other works, using 100 pieces of copy-move forgery data that they created, applying various attacks to those recordings, and studying 400 audio record files, Wang et al. used an entropy-based end-point voice activity detection algorithm to divide the audio into syllables. Afterward, they applied discrete cosine transform (DCT) to each syllable to extract features and the singular value decomposition (SVD) method to obtain singular vectors from the calculated DCT method. They also detected copy-move forgery by calculating the distance between two singular vectors [11].

Added to that, Yan et al. obtained the pitch sequence properties of each syllable of audio signals and subsequently calculated the similarity between those pitch sequences with PCCs and average difference methods for detecting copy-move forgeries. In their research, they generated 1000 copy-move forgery data, applied various attacks to those audio recordings, and studied a total of 3000 audio files [7].

By contrast, Liu et al. determined the voiced and unvoiced regions of audio signals by dividing the signals into syllables, segmented the signal to detect copy-move forgeries, applied DFT and Mel-frequency methods to each segment, and determined the features. They ultimately determined whether copy-move forgery could be detected in the audio signals according to the similarities between the segments using the PCC method. They also used 1000 pieces of forgery data in their study [17].

Huang et al. were the first to divide audio files into syllables to detect voiced and unvoiced regions in copy-move forgeries. After applying the DFT method to each segment, they compared the segments with each other [25]. Last, Xiao et al. also segmented audio signals to detect copy-move forgeries. In their work, they calculated the similarities between those segments with the fast convolution algorithm in order to detect copy-move forgeries in the signals [26].

3 Materials and methods

Our proposed end-to-end system was designed to detect whether a forgery has been performed on an audio signal based on the signal's Mel frequency characteristics and using ANNs. The proposed system consists of three steps: (1) detecting the audible and silent regions of an audio signal and segmenting it, (2) extracting the features based on Mel frequency from the segmented audio signal, and (3) classifying the extracted features with ANNs and, in turn, detecting audio copy-move forgery.

In this study, audio copy-move forgery detection, which was included in the audio forgery techniques given in Fig. 1, was carried out. The developed system, multiple modules were connected to detect forgery on audio signals. The data input module was the first module. The input signals were in.wav format and used as raw data therein. Input data were first labeled as forgery and non-forgery. Then, the segmentation of these signals takes place in the second module. In the second module, the generated dataset was divided into voiced and unvoiced segments using the yet another algorithm for pitch tracking (YAAPT). The YAAPT algorithm, which is based on both time and frequency domain, was adopted for segmenting the signal into voiced and unvoiced segments. In the third module, effective features were searched on each of the segments obtained from the voiced and unvoiced audio segments. MFCCs, ΔMFCCs, ΔΔMFCCs, and LPCs were obtained from all segments obtained as a result of the YAAPT and were recorded as features of the signals. Next, the features were extracted in the fourth module were subsequently divided into three parts as train, validation, and test data. The distribution of this data is 80% training and 20% test data. Another 20% of the training data were separated from the training data as validation data. In the classification phase in the fifth module, training and validation data were fed as input into the ANN model and trained with different hyperparameters. During the training of the model, 2801 train, 701 validation, and 876 test data were used. Then, in the sixth module, different models resulting from different ANN architectures are recorded. Last, in the seventh module, test data were given as input to the models obtained, and finally, in the eighth module, the performance analysis of the system is performed according to these data. The developed models and training results are shown in Tables 3–11, Fig. 2.

3.1 Yet another algorithm for pitch tracking (YAAPT)

Pitch tracking algorithms in the frequency domain are also widely used in the time domain. The YAAPT, based on both the time and frequency domain, is a pitch detection algorithm using the normalized cross-correlation function. Developed for high-quality voices and telephone conversations, it extracts the local maximum of the normalized cross-correlation function from an audio signal. The YAAPT is a very powerful algorithm because it extracts the fundamental frequency from audio signals with high accuracy. It consists of four steps [7, 20, 27–29]:

1. *Audio signal preprocessing* In preprocessing, an audio signal's multiple versions are created for the nonlinear processing of the signal. Nonlinear absolute value processing on audio signals can outperform most of method in terms of F0 tracking because it makes the fundamental frequency more apparent.
2. *F0 track calculation from signals using spectrum analysis* F0 traces are calculated using spectral harmonic correlations from nonlinear processed spectra of audio signals and dynamic programming. In this step, the pitch sequence from the processed audio is extracted as well as the original audio. The formula for spectral harmonic correlation (SHC) is given in Eq. 1:

$$\mathrm{SHC}(t,\ f) = \sum_{f'=-W_{\mathrm{len}}/2}^{W_{\mathrm{len}}/2} \prod_{k=1}^{H_{\mathrm{num}}+1} S\left(t,\ kf + f_n'\right) \tag{1}$$

in which $S(t, f)$ represents the short-time Fourier transform of the t frame at frequency f, H_{num} is the harmonic number, and W_{len} is the spectral window length. The NLFER is used to determine voiced and unvoiced segments. The NLFER has high-frequency values for voiced regions and low-frequency values for unvoiced regions. The NLFER formula is given in Eq. 2:

$$\mathrm{NLFER}(t) = \frac{\sum_{f=2\times F_{0\min}}^{F_{0\max}} S(t,\ f)}{\frac{1}{M}\sum_{t=1}^{\mathrm{M}} \sum_{f=2\times F_{0\min}}^{F_{0\max}} S(t,\ f)} \tag{2}$$

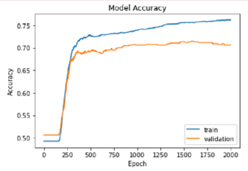
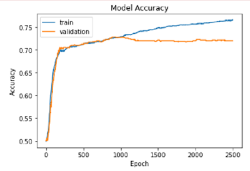
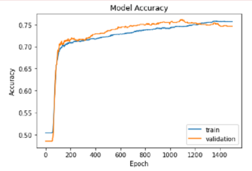
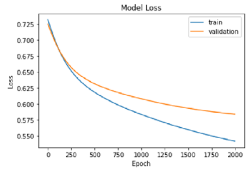
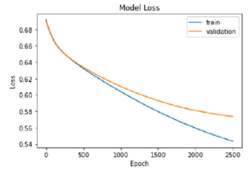
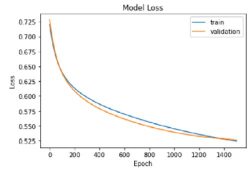
Table 1 Number of layers, neurons, and activation functions of the models

Model	Layers	No. of neurons	Activation Function	Optimizer Algorithm	Loss Function
M1	Layer1, Layer2, Output	8,4,1	Sigmoid	AdaDelta	binary_crossentropy
M2	Layer1, Layer2, Output	4,8,1	Sigmoid	AdaDelta	binary_crossentropy
M3	Layer1, Layer2, Output	16,8,1	Sigmoid	AdaDelta	binary_crossentropy

Table 2 Results of the MFCC method

Model	Epoch	Training accuracy	Validation accuracy	Test accuracy	Training accuracy	Validation accuracy	Test accuracy	Training accuracy	Validation accuracy	Test accuracy
		Batch size = 4			Batch size = 8			Batch size = 16		
M1	500	73.15	70.76	70.55	71.80	69.47	67.01	50.66	47.65	49.77
	1000	74.26	73.61	71.80	73.37	71.04	73.74	72.05	73.18	73.63
	1500	76.29	74.75	*73.63*	73.97	72.90	73.63	73.58	71.47	70.89
	2000	77.44	72.33	73.97	76.19	70.61	75.00	75.01	71.61	70.66
	2500	78.19	76.03	73.29	76.58	76.03	72.60	74.08	72.18	74.20
M2	500	71.58	72.04	74.09	71.58	69.04	69.29	50.16	49.93	49.54
	1000	74.01	74.32	72.60	71.26	72.90	71.80	72.47	69.33	71.80
	1500	75.54	72.18	*74.32*	74.58	74.47	73.40	72.94	71.90	72.95
	2000	76.90	72.47	72.72	76.01	72.61	74.43	75.22	72.04	71.12
	2500	77.08	73.18	71.92	76.62	72.04	74.43	74.83	73.61	74.77
M3	500	72.08	73.47	73.52	72.22	71.90	72.49	71.58	72.33	72.03
	1000	76.37	71.04	74.89	74.72	72.18	75.11	73.15	70.04	70.78
	1500	75.69	74.61	*75.34*	75.87	74.04	71.46	74.33	72.75	71.58
	2000	77.90	72.75	73.63	76.12	75.04	75.11	74.04	74.89	72.26
	2500	77.69	72.61	73.17	77.54	74.04	73.97	76.37	73.32	73.74

Table 3 Accuracy and loss graphs for the MFCC method

	M1 Model Batch size 8	M2 Model Batch size 16	M3 Model Batch size 4
Accuracy Graphs			
Loss Graphs			

in which $S(t ,f)$ represents the short-time Fourier transform of the t frame at frequency f and M is total number of frames.

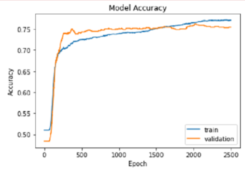
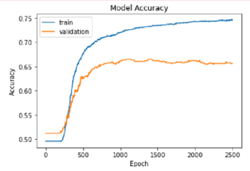
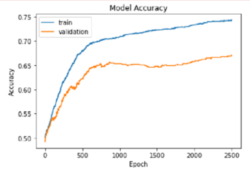
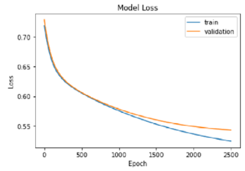
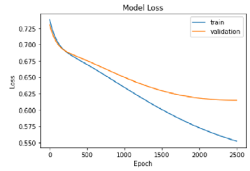
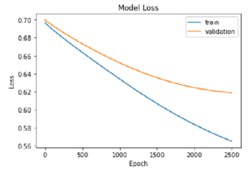
3. F0 candidate estimation F0 candidates are determined by normalized cross-correlation (NCCF) from the original and nonlinear processed signals. One of the greatest advantages of identifying F0 candidates with NCCF is that it can readily track rapid changes in pitch and amplitude. The NCCF formula is given in Eq. 3, the *ek* formula in Eq. 4, and the e0 formula in Eq. 5.

$$\mathrm{NCCF}(k) = \frac{1}{\sqrt{e_0 e_k}} \sum_{m=0}^{\mathrm{M-K\_max}} s(m)s(m+k)) \tag{3}$$

Table 4 Results of the ΔMFCC method

Model	Epoch	Training accuracy	Validation accuracy	Test accuracy	Training accuracy	Validation accuracy	Test accuracy	Training accuracy	Validation accuracy	Test accuracy
		Batch size = 4			Batch size = 8			Batch size = 16		
M1	500	61.69	61.34	62.33	49.88	51.07	49.54	52.02	50.07	50.80
	1000	66.90	67.19	65.30	70.26	66.76	70.43	51.91	53.21	52.51
	1500	72.33	71.90	70.55	72.08	69.33	71.12	64.62	66.19	63.47
	2000	74.15	69.76	68.26	73.33	66.62	71.00	72.19	73.47	70.09
	2500	73.55	69.90	72.26	77.08	75.46	73.97	73.08	70.04	71.00
M2	500	66.15	64.48	64.84	54.37	52.64	52.51	56.66	57.49	56.74
	1000	71.22	68.90	70.32	68.30	65.91	66.55	63.62	65.91	65.07
	1500	71.97	66.90	67.81	72.65	68.33	68.04	70.76	70.47	67.81
	2000	74.37	69.33	68.49	72.65	70.90	69.63	71.90	64.19	67.81
	2500	74.65	65.62	70.21	73.51	69.33	69.86	73.40	69.47	68.61
M3	500	71.51	67.19	66.67	68.51	65.05	62.67	59.51	53.64	55.02
	1000	73.08	68.47	70.21	71.51	70.33	68.15	70.97	69.33	67.24
	1500	74.47	66.90	68.26	73.33	70.04	67.69	71.65	67.48	68.72
	2000	74.72	68.47	68.38	73.90	70.04	71.80	72.55	70.76	67.47
	2500	75.62	68.62	69.18	74.76	68.19	72.83	74.29	67.05	72.26

Table 5 Accuracy and loss graphs for the ΔMFCC method

	M1 Model Batch size 8	M2 Model Batch size 4	M3 Model Batch size 16
Accuracy Graphs			
Loss Graphs			

$$e_k = \sum_{m=0}^{\text{k+M−K\_max}} s^2(m) \tag{4}$$

$$e_0 = \sum_{m=0}^{\text{k+M−K\_min}} s^2(m) \tag{5}$$

in which N is the frame length of the signal and K_min and K_max represent the hysteresis values used to fit the step's subtraction range.

4. F0 determination F0s are calculated by applying the dynamic programming technique considering the information obtained from Steps 1 and 2.

3.2 Mel frequency cepstral coefficients (MFCCs), ΔMFCCs, and ΔΔMFCCs

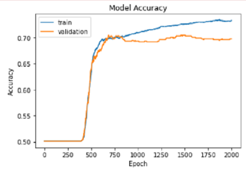
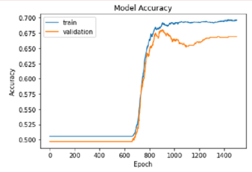
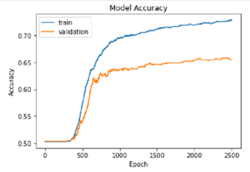
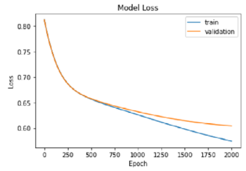
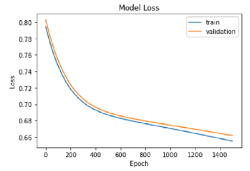
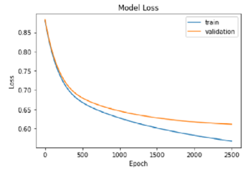
Using MFCC audio signals is an effective method of feature extraction frequently used in fields such as person detection, gender detection, and emotion detection. MFCCs have two types of filters that show linearity at frequencies below 1 kHz and logarithmic property at frequencies above 1 kHz [30, 31]. Audio signals are expressed in the Mel frequency scale in order to find the phonetic features in speech signals [32]. The formula used to convert the frequency value of the signals to the Mel frequency scale is given in Eq. 6, while the Δ_{Mel} frequency is given in Eq. 7 [33–35]:

$$f_{\text{mel}} = 1125\ln\left(1 + \frac{f}{700}\right) \tag{6}$$

Table 6 Results of the ΔΔMFCC method

Model	Epoch	Training accuracy	Validation accuracy	Test accuracy	Training accuracy	Validation accuracy	Test accuracy	Training accuracy	Validation accuracy	Test accuracy
		Batch size = 4			Batch size = 8			Batch size = 16		
M1	500	68.69	65.91	68.04	69.08	64.91	67.92	49.88	50.50	50.00
	1000	72.33	65.05	70.09	69.90	70.19	69.75	67.58	64.48	68.38
	1500	71.05	69.76	70.21	71.47	68.19	70.78	69.65	71.33	69.06
	2000	73.30	69.76	71.00	73.90	71.90	68.84	71.62	67.05	71.35
	2500	75.33	68.90	67.69	73.94	68.05	70.55	71.80	67.62	70.66
M2	500	68.83	62.05	67.12	68.08	66.05	67.92	65.05	64.62	67.01
	1000	70.44	67.33	68.04	68.37	66.62	69.63	63.30	60.91	60.50
	1500	71.33	68.19	67.24	69.62	66.90	69.86	69.26	68.62	65.30
	2000	73.40	67.76	69.41	71.19	69.33	69.75	69.58	71.18	67.12
	2500	73.15	67.05	69.52	72.76	70.04	68.15	71.19	68.62	67.12
M3	500	69.62	67.48	69.29	68.55	67.19	65.30	69.05	67.90	65.30
	1000	71.94	69.47	69.52	70.87	68.47	69.06	69.40	64.62	68.72
	1500	73.30	70.33	68.61	71.80	70.61	70.32	71.19	68.62	67.47
	2000	74.58	69.19	70.66	71.94	70.33	68.95	72.40	70.61	69.52
	2500	75.19	70.19	67.12	75.29	68.76	67.92	72.83	65.48	72.37

Table 7 Accuracy and loss graphs for the ΔΔMFCC method

	M1 Model **Batch size 4**	**M2 Model** **Batch size 8**	**M3 Model** **Batch size 16**
Accuracy Graphs			
Loss Graphs			

in which *f*_mel is the Mel frequency, and *f* is the frequency in Hertz. ΔMFCCs are obtained with the first-degree derivative of MFCCs, and ΔΔMFCCs are obtained with the second-degree derivative [36]; and

$$\text{delta}_t = \frac{\sum_{\tau=1}^{N} \tau (C_{t+\tau} - C_{t-\tau})}{2 * \sum_{\tau=1}^{N} \tau^2} \quad (7)$$

in which *delta* represents the ΔMFCCs and C_t the MFCCs.

Even the best classification algorithms can give weak results if the feature extraction methods are poorly defined [37]. Although MFCCs are static coefficients, because audio signals contain dynamic as well as static information, ΔMFCCs and ΔΔMFCCs are used to obtain dynamic information in the signals and reveal the temporal variability of the signals. Such features from audio signals play an important role in the classification phase. The most important purpose of feature extraction is to obtain various acoustic properties of audio signals [38]. The time domain, frequency domain, and time–frequency domains are used to define those features [39].

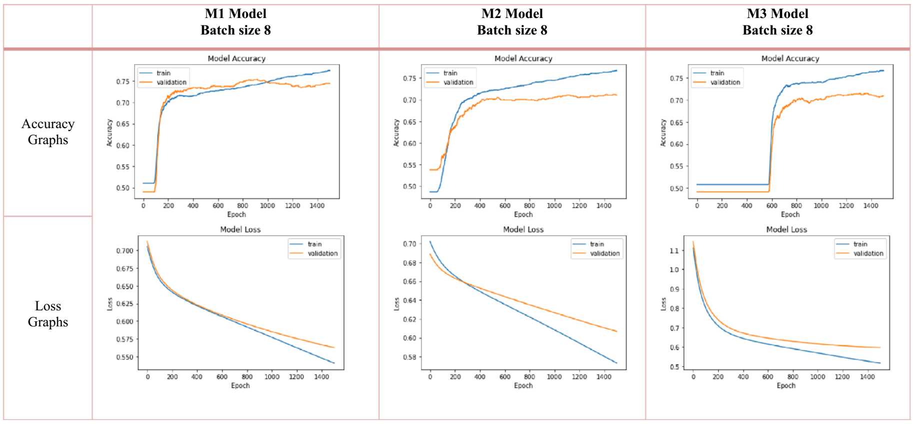
3.3 Linear prediction coefficients(LPCs*)*

In our study, LPCs were used to extract features from audio signals. Those coefficients are a powerful, efficient feature extraction method widely used in audio processing, and using LPCs, the time domain spectral features of audio data can be analyzed in a robust way [40]. LPCs have many advantages, including source–filter separation, orthogonality and compactness [41]. To LPCs, the prediction coefficient was first calculated by converting the 18-band Bark frequency

Table 8 Results of the MFCC + ΔMFCC + ΔΔMFCC method

Model	Epoch	Training accuracy	Validation accuracy	Test accuracy	Training accuracy	Validation accuracy	Test accuracy	Training accuracy	Validation accuracy	Test accuracy
		Batch size = 4			Batch size = 8			Batch size = 16		
M1	500	74.37	70.90	71.80	71.22	72.18	70.78	55.59	51.07	54.68
	1000	76.69	69.90	73.52	73.51	74.32	71.12	73.19	69.33	72.03
	1500	79.54	74.89	74.43	77.47	74.47	76.48	76.97	70.76	70.55
	2000	82.01	75.18	73.74	78.08	72.18	76.26	76.90	74.75	75.68
	2500	82.65	77.32	74.09	80.65	73.47	74.66	79.11	71.90	73.86
M2	500	73.05	71.61	70.66	65.12	62.05	63.01	49.59	51.21	50.34
	1000	75.47	72.33	74.43	74.65	74.32	73.29	72.44	73.89	72.95
	1500	77.47	71.75	75.00	76.72	71.04	75.23	75.90	73.47	70.55
	2000	80.86	73.75	73.52	79.90	74.32	74.89	77.08	73.47	74.09
	2500	82.93	72.04	72.49	81.29	71.75	73.06	77.83	76.32	71.69
M3	500	74.6	72.33	74.66	72.72	71.04	72.60	73.40	70.76	69.29
	1000	79.33	72.47	73.97	76.40	74.32	73.52	74.37	73.75	71.92
	1500	82.65	73.47	72.72	76.62	70.90	75.57	77.65	73.47	71.92
	2000	83.65	77.46	71.46	81.94	72.61	73.86	80.33	72.18	74.20
	2500	86.40	73.61	72.26	81.51	76.18	75.00	78.69	75.18	71.46

Table 9 Accuracy and loss graphs for the MFCC + ΔMFCC + ΔΔMFCC method



cepstrum into a linear frequency power spectral density. Second, the calculated spectral densities were converted into an autocorrelation by applying an FFT. Third and last, the Levinson–Durbin algorithm was used to calculate the predictor from the obtained autocorrelation [42, 43].

The LPC formula is given in Eq. 8 [40]:

$$\hat{y}(n) = \sum_{k=1}^{n} p_i y(n-k) \pm e(n) \tag{8}$$

in which n is the number of predictive samples, $x(n)$ and $x(n - 1)$ are the present and previous voice sample, p_i is the prediction factor, and $e(n)$ is the prediction error is given in Eq. 9 [40]:

$$e(n) = y(n) - \hat{y}(n) \tag{9}$$

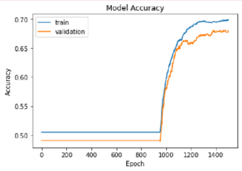
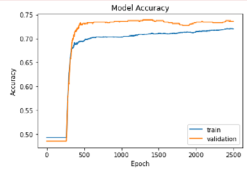
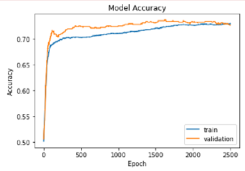
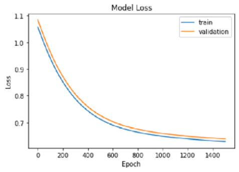
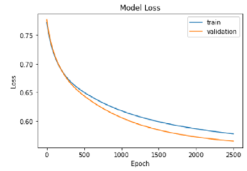
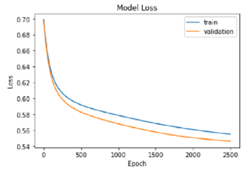
3.4 Artificial neural networks (ANNs)

ANNs are general structures that can create new information by learning, using the information automatically, and imitating the functioning of the human brain, thanks to neurons, the basic unit of the brain. ANNs represent a powerful mathematical model for finding complex relationships between inputs and outputs [44–46]. Used in many fields such as facial recognition, pupillary movement, handwriting recognition, heart arrhythmia diagnosis, robotics, identifying spam mail, and classifying financial data, ANNs have also played an important role in audio signal processing [47]. In general, ANNs constitute a highly robust method of adjusting classification and pattern recognition problems [48]. A neural network is

Table 10 Results of using the LPC method

Model	Epoch	Training accuracy	Validation accuracy	Test accuracy	Training accuracy	Validation accuracy	Test accuracy	Training accuracy	Validation accuracy	Test accuracy
		Batch size = 4			Batch size = 8			Batch size = 16		
M1	500	70.19	69.90	69.63	49.91	52.35	48.40	65.55	64.62	66.32
	1000	71.37	69.47	72.26	71.58	69.04	72.49	69.76	73.32	70.55
	1500	71.76	71.18	*72.49*	69.87	68.05	74.43	70.83	69.04	69.63
	2000	73.90	70.61	72.26	71.58	72.33	70.66	71.90	67.05	71.12
	2500	72.90	73.89	72.15	71.33	72.61	71.23	70.83	74.04	69.75
M2	500	69.76	71.18	72.15	49.95	50.36	49.89	50.45	53.21	46.00
	1000	72.40	68.47	68.49	71.15	69.47	69.18	71.05	68.47	71.00
	1500	71.55	72.47	*71.58*	71.37	70.33	70.32	70.72	69.76	71.80
	2000	71.55	74.47	69.63	70.62	74.89	71.23	71.51	69.47	69.18
	2500	72.05	73.61	74.77	72.94	71.33	68.38	71.30	70.19	71.23
M3	500	71.76	68.62	68.38	69.62	70.76	71.80	70.65	70.33	70.55
	1000	71.65	70.90	71.12	71.69	71.47	69.98	71.51	69.19	69.41
	1500	72.80	71.04	72.83	72.76	71.61	69.75	71.01	70.76	70.89
	2000	73.51	75.32	71.58	71.55	73.75	71.46	71.47	72.47	69.98
	2500	72.90	72.61	74.09	73.05	71.18	73.74	71.90	73.18	71.46

Table 11 Accuracy and loss graphs for the LPC method

	M1 Model Batch size 8	M2 Model Batch size 4	M3 Model Batch size 4
Accuracy Graphs			
Loss Graphs			

expressed in Eq. 10 [49]:

$$y_j^n = f\left(\sum_{i=1} w_{ji}^n x_i^{n-1} + \beta_j^n\right) \quad (10)$$

in which x the input neuron, w is the weight from artificial neuron, i in the $(n-1)$th layer to neuron j in the nth layer, β is a bias, and $f(x)$ is an activation function.

When developing an ANN model, the number of hidden layers, the number of neurons, and various hyperparameters in each hidden layer play a significant role [50]. A general neural network structure basically consists of three layers: the input layer, the hidden layer(s), and output layer. In the input and output layers, the input and output variables are determined. In the hidden layer(s), by contrast, the relationship between the inputs and outputs is determined. In an ANN model, the number of layers and neurons are determined in relation to the complexity of the data used. The learning phase in the ANN model is based on gradient transformation, in which the weights and threshold values at each node are optimized, and the error between the predicted and desired values is calculated and minimized as much as possible (Fig. 3).

4 Experimental results

4.1 Dataset

The TIMIT dataset contains 630 speaker files, ranging in length from 2 to 6 s, 438 of which depict male speakers

Table 12 Comparison of methods

Study	Method (s)	Decision system	Dataset
Su et al. (2020) [24]	CQCC, sliding window	MSE compared with the threshold, PCCs	Chinese speech and LibriSpeech dataset 500 tampered, 500 original audio files
Li et al. (2019) [20]	Pitch feature	Pearson correlation coefficient Compared with the threshold	500 audio files
Yan et al. (2019) [16]	Pitch feature and formant feature	Dynamic time warping Compared with the threshold	4000 different words from the TIMIT dataset and the *WSJ* audio database
Huang et al. (2020) [25]	DFT	Compared of each segment Compared with the threshold	–
Xie et al. (2018) [21]	Gammatone feature, MFCCs, pitch feature, and DFT	C4.5, decision tree, PCCs, and AD Compared with the threshold	Their own dataset 1,000 copy-move forgery and 1,000 audio files
Imran et al. (2017) [8]	1D LBP	MSE compared with each other Energy ratio used to compare histograms	King Saud University Arabic Speech Database
Wang et al. (2017) [11]	DCT and SVD	Distance between any two singular vectors calculated and compared with the threshold	Their own dataset 100 normal and 100 copy-move audio files
Liu et al. (2017) [17]	DFT and MFCCs	PCCs Compared with the threshold	Their own dataset 1,000 digital audio files
Yan et al. (2015) [7]	Pitch feature	PCCs and AD Compared with the threshold	Their own dataset 1000 tampered audios
Xiao et al. (2014) [26]	–	Fast convolution algorithm Compared with the threshold	–
Our method	MFCCs, ΔMFCCs, ΔΔMFCCs, MFCC + ΔMFCC + ΔΔMFCCs, and LPCs	Artificial neural network No threshold	4378 audio files from the TIMIT database

and 192 of which depict female speakers. The sampling frequency of the files is 16 kHz [51]. In our study, a copy-move forgery database was generated using 4378 audio files from the TIMIT dataset—2189 forgery audio record files and 2189 original audio recordings—varying in length from 4 to 6 s. The third second of each audio file was copied to the part after the end of the first second in the same file [22]. Thus, the third second of a 5 s audio file was copied to the part after the first second, such that the length of the 5 s audio recording after the copy-move forgery was 6 s total, meaning that a 1 s copy-move forgery process was generated on each audio file (Fig. 4).

4.2 Experimental setup

In our previous study [22], we analyzed the correlation between features obtained from MFCCs and ΔMFCCs. We determined whether copy-move forgery had occurred in the audio signal according to that correlation value by determining a certain threshold value used in relation to the correlation method. In another previous study [23], we detected copy-move forgery in audio signals using LPC features. We used PCCs to calculate the similarity between the features, and by comparing the similarity between those coefficients according to a determined threshold value, we detected copy-move forgery. In our study presented here, by contrast, we determined audio forgery by using a classification algorithm without any threshold value. We also used a larger dataset, added different feature extraction methods, and analyzed results for multiple features.

In our study, the method based on MFCCs, LPCs and ANNs was used to detect copy-move forgery in audio signals in 4378 audio recordings: 2189 of original audio and 2189 of copy-move forgery data. The proposed MFCC method, based on the frequency domain, produces more stable, more reliable results because it is less affected by the recorded environment, recording device, and features that vary from individual to individual. Moreover, because we considered temporal variation, the MFCC, ΔMFCC, and ΔΔMFCC methods were also examined and used. Our study consisted of three stages:

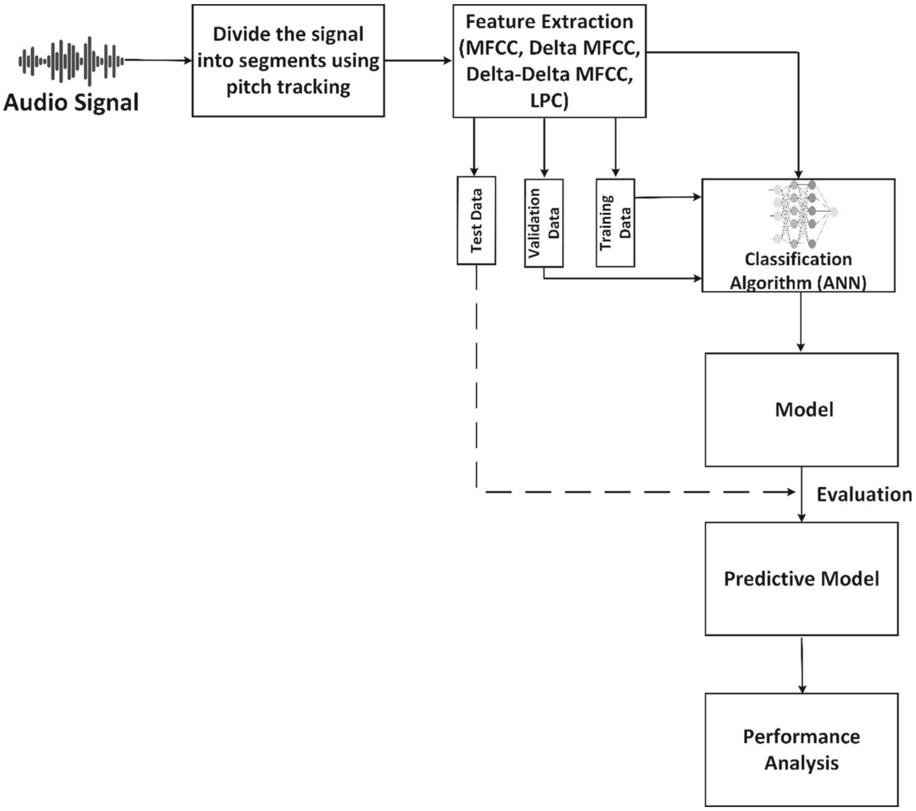

Audio Signal
Divide the signal into segments using pitch tracking
Feature Extraction (MFCC, Delta MFCC, Delta-Delta MFCC, LPC)
Test Data
Validation Data
Training Data
Classification Algorithm (ANN)
Model
Evaluation
Predictive Model
Performance Analysis

Fig. 2 A block diagram of our proposed method

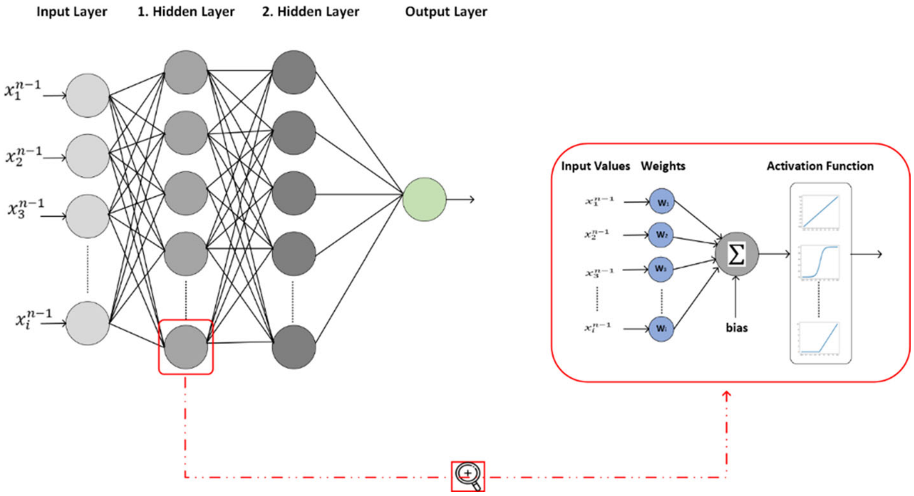


Fig. 3 A general neural network structure

1. *Preprocessing* Pitch tracking in audio signals is challenging. The most important reasons validated spectral characteristics of an audio signal change even in very small time intervals around milliseconds. The rapidity of such changes makes F0 tracking far more difficult than otherwise [29]. By contrast, the YAAPT can divide an audio signal into voiced and unvoiced regions for segmentation in our study because it can obtain the F0 fundamental frequency from audio signals with high accuracy. Figure 5a shows the waveform of the audio signal, (b) the pitch sequence, and (c) the combined representation of the audio signal and the pitch sequence. The YAAPT method was used as a pitch tracking algorithm to determine the pitch/fundamental frequency (F0) of the signal. The YAAPT algorithm was a signal processing model that determines the F0s by controlling the frame length, FFT length, the space between each frame, and the dynamic programming weighting factor for voiced/unvoiced transitions. Then, in the working principle of this algorithm, multiple versions of the signal were generated by applying a nonlinear preprocessing step to the audio signal. Next, spectrum analysis of the signal was performed to obtain pitch characteristics from these signals. The spectral correlation method was

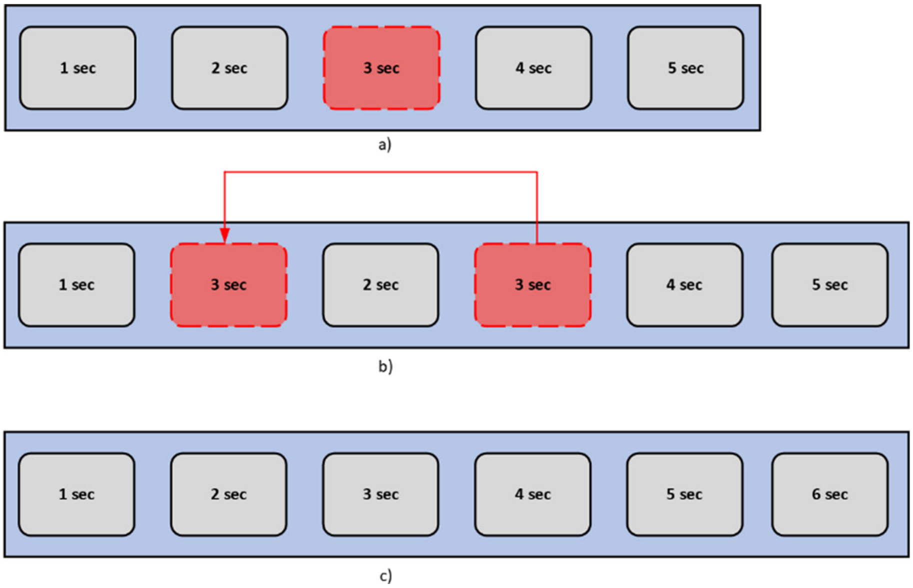


Fig. 4 **a** Original audio signal length, **b** situation of copy-move forgery to the audio signal, and **c** length of the audio signal after copy-move forgery

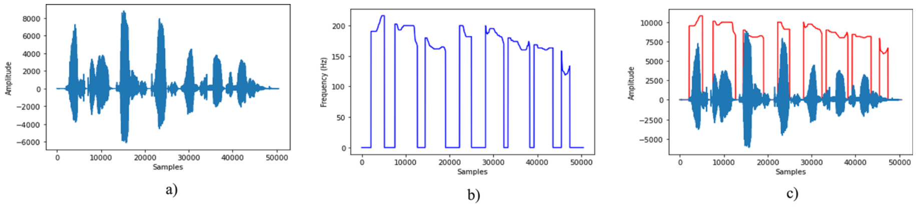


Fig. 5 **a** Waveform of the audio signal, **b** pitch sequence, and **c** combined representation of the audio signal and pitch sequence

used to extract pitch information from the audio signal. With the SHC method, the correlation value was calculated by examining the relationships between a frequency and a selected number of multiples of this frequency. When these relationships were similar, the correlation value is close to 1. When the relationships are weak, the correlation values were close to 0. The normalized low-frequency energy ratio (NLFER) method was used to identify the voiced and unvoiced segments in the audio signal. After this method, the voiced segments had high amplitude and the unvoiced segments had low amplitude values. In the next stage, the normalized cross-correlation function (NCCF) was used to obtain frequency information. At this stage, the NCCF method was used to select pitch candidates by calculating possible F0 candidates from the original and nonlinearly processed versions of the audio signal. In the final stage, a dynamic programming technique determines the best F0 among the candidate F0s. The YAAPT algorithm is used to separate the audio signal into its voiced and unvoiced segments. In the study, the length of each frame to be analyzed was 35 ms, the space between the analyzed frames was 10 ms, the FFT length was based on 8192 samples, and the normalized low frequency energy ratio was 0.75. To each segment, MFCCs, ΔMFCCs, ΔΔMFCCs, and LPCs were obtained from each segment. An audio signal from the TIMIT dataset (Fig. 5).

2. *Feature extraction* MFCCs were calculated by applying the Mel frequency transform to each segment during feature extraction from audio signals. Thus, a matrix of size $X \times 13$ was obtained for each segment, in which X is the row count. The mean values of the rows in each matrix were calculated, and the mean MFCCs of 1×13 were obtained for each audio segment. Those coefficients served as the input of the ANN such that forgery in the audio signals could be detected. The most fitting model for our study was established by changing the number of layers, number of neurons, activation functions, and batch size in the ANN model. In our study, the window size was 0.025 s, WinStep = 0.01, the step length between consecutive windows was 0.01 s, and the number of filters in the filter bank was 26. LPCs were calculated using five coefficients for each segment from audio signals during feature extraction.
3. ANNs utilized to classify MFCCs in the classification phase (Fig. 6).

In our study, three models using default hyperparameters were used in the classification phase. The number of layers, neurons, and activation functions in the models used appear in Table 1.

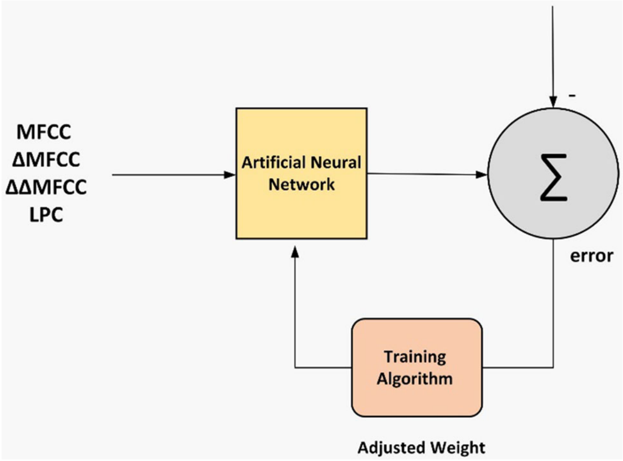


Fig. 6 Overview of the system in our study

4.3 Results

The training, validation, and test accuracies of the proposed method for each classification model were calculated, and the results concerning performance were compared.

The MFCC method's results are given in Table 2. Our performance metrics are given in Eq. 11:

$$\text{Accuracy}(\text{Acc}) = \frac{1}{N}\sum_{i}^{N} y_i == y_i^{\text{prediction}} \qquad (11)$$

in which y_i is the actual values, $y_i^{\text{prediction}}$ is the predicted values, and N is the number of items.

The MFCC method was trained with different numbers of epochs and batch sizes, and the best model for detecting copy-move forgery was determined. The best accuracy rate for MFCC features was calculated in the M3 model with 1500 epochs and a batch size of 4. The training accuracy of M3 was 75.69%, its validation accuracy was 74.61%, and its test accuracy was 75.34%. As shown in Table 2:

- In M1, the best accuracy rate was calculated with 2000 epochs and a batch size of 8. The training accuracy was 76.19%, the validation accuracy was 70.61%, and the test accuracy was 75.00%.
- In M2, the best accuracy rate was calculated with 2500 epochs and a batch size of 16. The training accuracy is 74.83%, the validation accuracy was 73.61%, and the test accuracy was 74.77%.
- In M3, the best accuracy rate was calculated with 1500 epochs and a batch size of 4. The training accuracy of the M3 model was 75.69%, the validation accuracy was 74.61%, and the test accuracy was 75.34%.
- In all models (i.e., M1, M2, and M3), when the batch size was 4, the best accuracy rates were obtained with 1500 epochs; when the batch size was 8, the best accuracy rates were obtained with 2000 epochs; and when the batch size was 16, the best accuracy rates were obtained with 2500 epochs. Thus, as the batch size increased, the number of epochs increased as well.

Table 3 presents graphs of the accuracy and loss of the best results in M1, M2, and M3 using the MFCC method.

The ΔMFCC method's results are given in Table 4.

The ΔMFCC method was also trained with different numbers of epochs numbers and batch sizes, and the best model for detecting copy-move forgery was determined. The best accuracy rate for ΔMFCC features was calculated in M1 with 2500 epochs and a batch size of 8. The training accuracy of M1 was 77.08%, its validation accuracy was 75.46%, and its test accuracy was 73.97%. As shown in Table 4:

- In M1, the best accuracy rate was calculated with 2500 epochs and a batch size of 8. The training accuracy was 77.08%, the validation accuracy was 75.46%, and the test accuracy was 73.97%.
- In M2, the best accuracy rate was calculated with 2500 epochs and a batch size of 4. The training accuracy was 74.65%, the validation accuracy was 65.62%, and the test accuracy was 70.21%.
- In M3, the best accuracy rate was calculated with 2500 epochs and a batch size of 16. The training accuracy was 74.29%, the validation accuracy was 67.05%, and the test accuracy was 72.26%.
- In all batch sizes for all models (i.e., M1, M2, and M3), the best accuracy rates were obtained with 2500 epochs. Thus, regardless of whether the batch size changed, the number of best epochs did not change.

Table 5 presents graphs of the accuracy and loss of the best results in M1, M2, and M3 using the ΔMFCC method.

The ΔΔMFCC method was also trained with different numbers of epochs and batch sizes, and the best model for detecting copy-move forgery was determined. The best accuracy rate for ΔΔMFCC features was calculated with M3 with 2500 epochs and a batch size of 16. The training accuracy of M3 was 72.83%, its validation accuracy was 65.48%, and its test accuracy was 72.37%. As shown in Table 6:

- In M1, the best accuracy rate was calculated with 2000 epochs and a batch size of 4. The training accuracy was 73.30%, the validation accuracy was 69.76%, and the test accuracy was 71.00%.
- In M2, the best accuracy rate was calculated with 1500 epochs and a batch size of 8. The training accuracy was 69.62%, the validation accuracy was 66.90%, and the test accuracy was 69.86%.

- In M3, the best accuracy rate was calculated with 2500 epochs and a batch size of 16. The training accuracy was 72.83%, the validation accuracy was 65.48%, and the test accuracy was 72.37%.
- In all models (i.e., M1, M2, and M3), when the batch size was 4, the best accuracy rates were obtained with 2000 epochs; when the batch size was 8, the best accuracy rates were obtained with 1500 epochs; and when the batch size was 16, the best rates were obtained with 2500 epochs. Thus, the increase in batch size is related to the increase in epochs.

The accuracy and loss graphs of the best results with M1, M2, and M3 using the $\Delta\Delta$MFCC method appear in Table 7.

Results obtained with the MFCC + ΔMFCC + $\Delta\Delta$MFCC method appear in Table 9.

Last, the MFCC + ΔMFCC + $\Delta\Delta$MFCC method was also trained with different numbers of epochs and batch sizes, and the best model for detecting copy-move forgery was determined. The best accuracy rate for MFCC + ΔMFCC + $\Delta\Delta$MFCC features was calculated with M1 with 1500 epochs and a batch size of 8. The training accuracy of M1 was 77.47%, its validation accuracy was 74.47%, and its test accuracy was 76.48%. As shown in Table 8:

- In M1, the best accuracy rate was calculated with 1500 epochs and a batch size of 8. The training accuracy was 77.47%, the validation accuracy was 74.47%, and the test accuracy was 76.48%.
- In M2, the best accuracy rate was calculated with 1500 epochs and a batch size of 8. The training accuracy was 76.72%, the validation accuracy was 71.04%, and the test accuracy was 75.23%.
- In M3, the best accuracy rate was calculated with 1500 epochs and a batch size of 8. The training accuracy of M3 was 76.62%, the validation accuracy was 70.90%, and the test accuracy was 75.57%.
- In all models (i.e., M1, M2, and M3), when the batch size was 4, the best accuracy rates were obtained with 1500 epochs; when the batch size was 8, the best accuracy rates were obtained with 1500 epochs; and when the batch size was 16, the best accuracy rates were obtained with 2000 epochs. Thus, the increase in epochs is related to the increase in batch size.

The accuracy and loss graphs of the best results from M1, M2, and M3 using the MFCC + ΔMFCC + $\Delta\Delta$MFCC method appear in Table 9.

The LPC method was also trained with different numbers of epochs and batch sizes, and the best model for detecting copy-move forgery was determined. The best accuracy rate for the LPC features was calculated with M3 with 2500 epochs and a batch size of 4. The training accuracy of M3 was 72.90%, its validation accuracy was 72.61%, and its test accuracy was 74.09%. As shown in Table 10:

- In M1, the best accuracy rate was calculated with 1500 epochs and a batch size of 8. The training accuracy was 69.87%, the validation accuracy was 68.05%, and the test accuracy was 74.43%.
- In M2, the best accuracy rate was calculated with 2500 epochs and a batch size of 4. The training accuracy was 72.05%, the validation accuracy was 73.61%, and the test accuracy was 74.77%.
- In M3, the best accuracy rate was calculated with 2500 epochs and a batch size of 4. The training accuracy was 72.90%, the validation accuracy was 72.61%, and the test accuracy was 74.09%.
- In all models (i.e., M1, M2, and M3), when the batch size was 4, the best accuracy rates were obtained with 2500 epochs; when the batch size was 8 and 16, however, the best accuracy rates changed depending on the model and number of epochs. Thus, as the batch size increased, so did the number of epochs.

The accuracy and loss graphs of the best results with M1, M2, and M3 using the LPC method appear in Table 11.

Table 12 compares the method of our proposed model with alternative methods in the literature.

Proposed method results are compared with CQCC, sliding window [24], pitch feature method [7, 20], pitch feature and formant feature methods [16], DFT method [25], gammatone feature, MFCCs, pitch feature and DFT [21], DCT method, SVD method [11], DFT method, MFCC methods results [17]. Results are given in Table 12. Most of research were based on a threshold value and statistical decision making methods. We propose a robust method that does not use a threshold and can work under different conditions. Various copy-move forgery detection experiments show that the proposed method was competitive with other state-of-the-art methods.

5 Discussion

Detecting audio copy-move forgery is one of the most challenging but important tasks in audio forensics. Even so, no reliable integrated system for detecting audio copy-move forgery exists, and studies based on neural networks for detecting copy-move forgery have been few and far between. In their studies, researchers have identified the voiced and unvoiced regions of audio recordings, divided the signals into segments, obtained various features from those signs, and, using mostly statistical methods, ultimately detected copy-move forgery in light of a certain threshold value. The disadvantage of that approach, however, is that the developed system depends on the threshold value, which itself depends on certain conditions. In other words, the system is not robust in different conditions and situations. The threshold value may differ depending on the length of the added region in the audio recordings, whether the audio recordings are noisy or not, the environment in which the recordings were made, the microphone used, and the speaker's gender. Those challenges have to be overcome in any system intended to work under all conditions. It is pivotal for a copy-move forgery detection system to give highly accurate, highly reliable results regardless of the noise, the environment, the voice recorder used, and the speaker, for the audio signals are required to be strong and stable in order to work. In our study, we developed a state-of-the-art method of detecting an audio copy-move forgery in a powerful, stable way that can be adapted to all kinds of variable conditions without the need for any threshold value. Furthermore, the accuracy of method seemed to generally relate to larger datasets. If the dataset is larger, then the performance increases. Our study was the first in which researchers have attended to the accuracy of validation. Compared with the articles presented in Table 12, our proposed method was developed based on the largest dataset. For those reasons, our paper can be expected to contribute to the literature.

Beyond that, our study has revealed that the epoch count and batch size are related in models training. However, considering the MFCC, ΔMFCC, ΔΔMFCC and LPC features, the results were directly affected by a derivation process. If by using a derivation on the MFCC feature, the accuracy results have decrease. When the features were used together, the models afforded far better results, even at small epochs. Thus, the greater the number of MFCC + ΔMFCC + ΔΔMFCC features, the higher the accuracy of the results.

6 Conclusions

Forgery detection in audio signals is one of the most examined topics in audio forensics. In our study, each voiced and unvoiced region in an audio signal was determined using the YAAPT algorithm. Feature vectors in those voiced and unvoiced regions were obtained by applying four feature extraction methods—the MFCC, ΔMFCC, ΔΔMFCC, and MFCC + ΔMFCC + ΔΔMFCC methods—and three ANN models were applied to each feature set obtained. According to results concerning comprehensive performance, the best results from the feature vector were calculated by using MFCCs, ΔMFCCs, and ΔΔMFCCs together in M1. To be specific, the best results were a 77.47% training rate, a 74.47% validation rate, and a 76.48% test accuracy rate.

Acknowledgements Our study was supported by the Scientific and Technological Research Council of Turkey (Grant No. 121E725).

Author contributions FA was involved in the conceptualization, method, software, validation, formal analysis, investigation, software, resources, and data curation. YB contributed to the software, validation, supervision, conceptualization, review, and editing.

Funding We received research funding from the Scientific and Technological Research Council of Turkey (Grant No. 121E725).

Data availability Data will be made available on reasonable request.

Declarations

Conflict of interest The authors declare no conflict of interest.

References

1. Khan, M.K., Zakariah, M., Malik, H., Choo, K.K.R.: A novel audio forensic data-set for digital multimedia forensics. Aust. J. Forensic Sci. **50**(5), 525–542 (2018). https://doi.org/10.1080/00450618.2017.1296186
2. Bourouis, S., Alroobaea, R., Alharbi, A.M., Andejany, M., Rubaiee, S.: Recent advances in digital multimedia tampering detection for forensics analysis. Symmetry **12**(11), 1811 (2020). https://doi.org/10.3390/sym12111811
3. Sunitha, K., Krishna, A.N., Prasad, B.G.: Copy-move tampering detection using keypoint based hybrid feature extraction and improved transformation model. Appl. Intell. **52**(13), 15405–15416 (2022)
4. Patel, R., Lad, K., Patel, M.: Study and investigation of video steganography over uncompressed and compressed domain: a comprehensive review. Multimedia Syst. **27**(5), 985–1024 (2021)
5. Kasapoğlu, B., Turgay, K.O.Ç.: Sentetik ve Dönüştürülmüş Konuşmaların Tespitinde Genlik ve Faz Tabanlı Spektral Özniteliklerin Kullanılması. Avrupa Bilim ve Teknoloji Dergisi, pp. 398–406. (2020). https://doi.org/10.31590/ejosat.780650
6. Javed, A., Malik, K.M., Irtaza, A., Malik, H.: Towards protecting cyber-physical and IoT systems from single-and multi-order voice spoofing attacks. Appl. Acoust. **183**, 108283 (2021). https://doi.org/10.1016/j.apacoust.2021.108283
7. Yan, Q., Yang, R., Huang, J.: Copy-move detection of audio recording with pitch similarity. In: 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 1782–1786). IEEE (2015)
8. Imran, M., Ali, Z., Bakhsh, S.T., Akram, S.: Blind detection of copy-move forgery in digital audio forensics. IEEE Access **5**, 12843–12855 (2017). https://doi.org/10.1109/ACCESS.2017.2717842
9. Wang, Z., Yang, Y., Zeng, C., Kong, S., Feng, S., Zhao, N.: Shallow and deep feature fusion for digital audio tampering detection. EURASIP J. Adv. Signal Process. **2022**(1), 1–20 (2022)
10. Maher, R.C.: Audio forensic examination. IEEE Signal Process. Mag. **26**(2), 84–94 (2009). https://doi.org/10.1109/MSP.2008.931080
11. Wang, F., Li, C., Tian, L.: An algorithm of detecting audio copy-move forgery based on DCT and SVD. In: 2017 IEEE 17th International Conference on Communication Technology (ICCT) (pp. 1652–1657). IEEE (2017)
12. Jadhav, S., Patole, R., Rege, P.: Audio splicing detection using convolutional neural network. In: 2019 10th International Conference

on Computing, Communication and Networking Technologies (ICCCNT) (pp. 1–5). IEEE (2019)
13. Chen, J., Xiang, S., Huang, H., Liu, W.: Detecting and locating digital audio forgeries based on singularity analysis with wavelet packet. Multimedia Tools Appl. **75**(4), 2303–2325 (2016). https://doi.org/10.1007/s11042-014-2406-3
14. Yang, R., Qu, Z., Huang, J.: Detecting digital audio forgeries by checking frame offsets. In: Proceedings of the 10th ACM Workshop on Multimedia and Security (pp. 21–26) (2008)
15. Gupta, S., Cho, S., Kuo, C.C.J.: Current developments and future trends in audio authentication. IEEE Multimedia **19**(1), 50–59 (2011). https://doi.org/10.1109/MMUL.2011.74
16. Yan, Q., Yang, R., Huang, J.: Robust copy-move detection of speech recording using similarities of pitch and formant. IEEE Trans. Inform. Forensics Secur. **14**(9), 2331–2341 (2019). https://doi.org/10.1109/TIFS.2019.2895965
17. Liu, Z., Lu, W.: Fast copy-move detection of digital audio. In: 2017 IEEE Second international conference on data science in cyberspace (DSC) (pp. 625–629). IEEE (2017)
18. Ali, Z., Imran, M., Alsulaiman, M.: An automatic digital audio authentication/forensics system. IEEE Access **5**, 2994–3007 (2017). https://doi.org/10.1109/ACCESS.2017.2672681
19. Bevinamarad, P.R., Shirldonkar, M.S.: Audio forgery detection techniques: present and past review. In: 2020 4th International Conference on Trends in Electronics and Informatics (ICOEI)(48184) (pp. 613–618). IEEE (2020)
20. Li, C., Sun, Y., Meng, X., Tian, L.: Homologous audio copy-move tampering detection method based on pitch. In: 2019 IEEE 19th International Conference on Communication Technology (ICCT) (pp. 530–534). IEEE (2019)
21. Xie, Z., Lu, W., Liu, X., Xue, Y., Yeung, Y.: Copy-move detection of digital audio based on multi-feature decision. J. Inform. Secur. Appl. **43**, 37–46 (2018). https://doi.org/10.1016/j.jisa.2018.10.003
22. Akdeniz, F., Becerikli, Y.: Detection of copy-move forgery in audio signal with mel frequency and delta-mel frequency kepstrum coefficients. In: 2021 Innovations in Intelligent Systems and Applications Conference (ASYU) (pp. 1–6). IEEE (2021)
23. Akdeniz, F., Becerikli, Y.: Linear prediction coefficients based copy-move forgery detection in audio signal. In: 2022 International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT) (pp. 770–773). IEEE (2022)
24. Su, Z., Li, M., Zhang, G., Wu, Q., Wang, Y.: Robust audio copy-move forgery detection on short forged slices using sliding window. J. Inform. Secur. Appl. **75**, 103507 (2023)
25. Huang, X., Liu, Z., Lu, W., Liu, H., Xiang, S.: Fast and effective copy-move detection of digital audio based on auto segment. In: Digital Forensics and Forensic Investigations: Breakthroughs in Research and Practice (pp. 127–142). IGI Global (2020). https://doi.org/10.4018/978-1-7998-3025-2.ch011
26. Xiao, J.N., Jia, Y.Z., Fu, E.D., Huang, Z., Li, Y., Shi, S.P.: Audio authenticity: duplicated audio segment detection in waveform audio file. J. Shanghai Jiaotong Univ. (Sci.) **19**(4), 392–397 (2014). https://doi.org/10.1007/s12204-014-1515-5
27. Kadiri, S.R., Yegnanarayana, B.: Estimation of fundamental frequency from singing voice using harmonics of impulse-like excitation source. In: Interspeech (pp. 2319–2323) (2018)
28. Zahorian, S.A., Hu, H.: A spectral/temporal method for robust fundamental frequency tracking. J. Acoust. Soc. Am. **123**(6), 4559–4571 (2008). https://doi.org/10.1121/1.2916590
29. Kasi, K.: Yet another algorithm for pitch tracking: (YAAPT) (Doctoral dissertation, Old Dominion University) (2002)
30. Muda, L., Begam, M., Elamvazuthi, I.: Voice recognition algorithms using mel frequency cepstral coefficient (MFCC) and dynamic time warping (DTW) techniques. arXiv preprint arXiv:1003.4083. (2010). https://doi.org/10.48550/arXiv.1003.4083
31. Ancilin, J., Milton, A.: Improved speech emotion recognition with Mel frequency magnitude coefficient. Appl. Acoust. **179**, 108046 (2021)
32. Hasan, M.R., Jamil, M., Rahman, M.G.R.M.S.: Speaker identification using mel frequency cepstral coefficients. Variations **1**(4), 565–568 (2004)
33. Das, P.P., Allayear, S.M., Amin, R., Rahman, Z.: Bangladeshi dialect recognition using Mel frequency cepstral coefficient, delta, delta-delta and Gaussian mixture model. In: 2016 Eighth International Conference on Advanced Computational Intelligence (ICACI) (pp. 359–364). IEEE (2016)
34. Hossan, M.A., Memon, S., Gregory, M.A.: A novel approach for MFCC feature extraction. In: 2010 4th International Conference on Signal Processing and Communication Systems (pp. 1–5). IEEE (2010)
35. Abo-Zahhad, M., Farrag, M., Abbas, S.N., Ahmed, S.M.: A comparative approach between cepstral features for human authentication using heart sounds. SIViP **10**(5), 843–851 (2016)
36. YÜCESOY, E.: MFKK Özniteliklerine Eklenen Logaritmik Enerji ve Delta Parametrelerinin Yaş ve Cinsiyet Sınıflandırma Üzerindeki Etkileri. J. Ins. Sci. Technol. **11**(1), 32–43 (2021). https://doi.org/10.21597/jist.772804
37. Akdeniz, F., Kayikcioglu, İ, Kayikcioglu, T.: Classification of cardiac arrhythmias using Zhao-Atlas-Marks time-frequency distribution. Multimedia Tools Appl. **80**(20), 30523–30537 (2021). https://doi.org/10.1007/s11042-021-10945-6
38. Gupta, S., Shukla, R.S., Shukla, R.K.: Weighted Mel frequency cepstral coefficient based feature extraction for automatic assessment of stuttered speech using Bi-directional LSTM. Indian J. Sci. Technol. **14**(5), 457–472 (2021). https://doi.org/10.17485/IJST/v14i5.2276
39. Abeysinghe, A., Fard, M., Jazar, R., Zambetta, F., Davy, J.: Mel frequency cepstral coefficient temporal feature integration for classifying squeak and rattle noise. J. Acoust. Soc. Am. **150**(1), 193–201 (2021). https://doi.org/10.1121/10.0005201
40. Prabakaran, D., Shyamala, R.: A review on performance of voice feature extraction techniques. In: 2019 3rd International Conference on Computing and Communications Technologies (ICCCT) (pp. 221–231). IEEE (2019)
41. Sharma, G., Umapathy, K., Krishnan, S.: Trends in audio signal feature extraction methods. Appl. Acoust. **158**, 107020 (2020)
42. Valin, J.M., Skoglund, J.: LPCNet: improving neural speech synthesis through linear prediction. In: ICASSP 2019–2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 5891–5895). IEEE (2019)
43. Juvela, L., Tsiaras, V., Bollepalli, B., Airaksinen, M., Yamagishi, J., Alku, P.: Speaker-independent raw waveform model for glottal excitation. arXiv preprint arXiv:1804.09593 (2018)
44. Siniscalchi, S.M., Svendsen, T., Lee, C.H.: An artificial neural network approach to automatic speech processing. Neurocomputing **140**, 326–338 (2014). https://doi.org/10.1016/j.neucom.2014.03.005
45. Güraksin, G.E.: Kalp seslerinin yapay sinir ağları ile sınıflandırılması (Master's thesis, Fen Bilimleri Enstitüsü).(2009)
46. Akdeniz, F., Becerikli, Y.: Performance comparison of support vector machine, k-nearest-neighbor, artificial neural networks, and recurrent neural networks in gender recognition from voice signals. In: 2019 3rd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT) (pp. 1–4). IEEE (2019)
47. Machado, T.J., Vieira Filho, J., de Oliveira, M.A.: Forensic speaker verification using ordinary least squares. Sensors **19**(20), 4385 (2019). https://doi.org/10.3390/s19204385

48. Tibdewal, M.N., Fate, R.R., Mahadevappa, M., Ray, A.K., Malokar, M.: Classification of artifactual EEG signal and detection of multiple eye movement artifact zones using novel time-amplitude algorithm. SIViP **11**(2), 333–340 (2017)
49. Wang, F., Chen, Z., Wu, C., Yang, Y.: Prediction on sound insulation properties of ultrafine glass wool mats with artificial neural networks. Appl. Acoust. **146**, 164–171 (2019). https://doi.org/10.1016/j.apacoust.2018.11.018
50. Kır Savaş, B., Becerikli, Y.: Behavior-based driver fatigue detection system with deep belief network. Neural Comput. Appl. (2022). https://doi.org/10.1007/s00521-022-07141-4
51. Garofolo, J.S., Lamel, L.F., Fisher, W.M., Fiscus, J.G., Pallett, D.S.: DARPA TIMIT acoustic-phonetic continous speech corpus CD-ROM. NIST speech disc 1–1.1. NASA STI/Recon Tech. Rep N **93**, 27403 (1993)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.