APSIPA Transactions on Signal and Information Processing, 2025, 14, e19

Original Paper

ASVSpoof 2021: Detecting Spoofed Utterances Through Hybrid Features

Ramesh K. Bhukya[1], Aditya Raj[2] and Anshul Kumar[3]*

[1]*Department of Electronics and Communication Engineering, Indian Institute of Information Technology, Allahabad, India*

[2]*Department of Information Technology-Business Informatics, Indian Institute of Information Technology, Allahabad, India*

[3]*Department of Electronics and Communication Engineering, Birla Institute of Technology Mesra, India*

ABSTRACT

ASVSpoof is a set of challenges intended to advance research into the spoofing risks to *automated speaker verification* (ASV) systems. Giving a false speech signal that mimics the characteristics of a real speech signal is a common technique for tricking an ASV system. Spoofing is the practice of impersonating another speaker. ASVspoof uses three assessment measures, *Logical Access* (LA), *Physical Access* (PA) and *DeepFake* (DF), to assess the effectiveness of spoofing defences developed for ASV systems. In this study, we used the k-Nearest Neighbour (k-NN), *Support Vector Machine* (SVM), *Random Forest* (RF), *Gradient Boosting* (GB), *AdaBoost*, *XGBoost*, and *Multi-Layer Perceptron* (MLP) are *Machine Learning* (ML) models. DNN-single, DNN-CNN, DNN-convLSTM, and DNN-BiLSTM are *Deep Learning* (DL) models to assess the ASVspoof on the ASVspoof2021 datasets. DL entails the process of transforming manually crafted *feature vectors* (FVs) into more extensive, dense FVs via matrix multiplication. A DL model's architecture may be modified to fit the particular application, offering flexibility in terms of the number of layers, hidden layer dimensions, utilized transformation functions, and

*Corresponding author: Ramesh K. Bhukya, rkbhukya@iiita.ac.in.

Received 04 November 2024; revised 23 December 2024; accepted 25 January 2025
ISSN 2048-7703; DOI 10.1561/116.20250026

selected loss functions. In this study, we created specialized DL architectures that were suited to the ASVSpoof dataset, assuring both computational and temporal effectiveness. With the above algorithm, the ML models have an accuracy of 90% for k-NN, 96% for SVM, 95% for RF, 95% for GB, 92% for AdaBoost, 96% for XGBoost, and 95% for MLP. When it is applied to the DL models, it shows more than 99% accuracy in DNN-Single, DNN-CNN, DNN-convLSTM, and DNN-BiLSTM. It demonstrates that the DL algorithm on ASVspoof 2021 data shows more accuracy.

Keywords: Automatic Speaker Verification, ASVspoof, Logical Access, Physical Access, DeepFake, Hybrid Features, Countermeasure, Machine Learning, Deep Learning.

1 Introduction

The *automatic speaker verification* (ASV) is a biometric technique that aims to verify the claimed identity of a speaker using their voice characteristics [11]. It includes analysing speech signals using algorithms and statistical models and then making judgments on the veracity of the speaker. ASV has made considerable strides over the years, transitioning from conventional methodologies to cutting-edge strategies that make use of *machine learning* (ML) and *deep learning* (DL) techniques. An ASV-protected resource, service, or device may be vulnerable to spoofing attacks that grant an attacker unauthorised access [59]. Spoofing poses a serious and unacceptable threat. The ASVSpoof initiative has taken the lead in the effort to build spoofing countermeasures, auxiliary systems that attempt to defend ASV technology by automatically detecting and deflecting spoofing assaults since the first special session on anti-spoofing was held in 2013. However, the crucial component in synthetic speech identification is the artefact left by data forgery, which could not include any semantic information [81]. It was created in response to the problem of voice signals that may be faked or otherwise altered to fool ASV systems, allowing for impersonation or unauthorised access. The performance of ASV systems on ASVspoof2013 [37] is typically evaluated using metrics such as the *equal error rate* (EER) and the *minimum normalised detection cost function* (minDCF) [66]. These metrics measure the ability of ASV systems to distinguish between genuine and spoofed speech, with lower values indicating better performance [23]. Secondly, the 2019 version investigated replay assaults utilising a far more controlled assessment setup, including simulated replay attacks and meticulously regulated acoustic settings. Unsupervised generative models, such as the SVM, are frequently used to represent the probability distribution of

audio characteristics [10]. Such a link, however, is made without channel-wise priority by adding features directly to one another in feature groups [52]. The goal of ASVspoof 2019 [85] was to determine whether recent advancements in *voice conversion* (VC) and *speech synthesis* (SS) technologies pose a greater threat to ASV reliability. For instance, waveform modelling techniques based on neural networks may create artificial speech that is perceptually identical to genuine speech.

Other speech processing techniques, such as VC and SS, are becoming more widely accessible and of higher quality, and they have begun to pose a serious challenge to ASV systems [33]. The need for technologies that can automatically assess the integrity of those materials grows along with the quality of generative algorithms. A good replay detection system should be capable of detecting both known and unknown circumstances that are present in the challenge [34]. It is becoming increasingly challenging to spot spoofed data since falsified pictures, videos, and audio just keep getting more realistic thanks to emerging technologies like *DeepFakes* (DFs). Additionally, due to the increasing use of recording devices (such as smart speakers and smartphones), the voice inputs for ASV systems are subject to a variety of channel variations. The earlier methods used extremely sophisticated classifiers to manually seek different spectrum properties or learnt features in that way [70]. Each data set contains spoof speech produced by various SS, VC, and hybrid techniques [77]. The ASV performance will suffer from this channel mismatch between the samples. The spoof detection system will also be confused since the channel distortion may mask the spectrum artefacts produced by the spoof creation process (such as VC or SS). With several spoofing assaults, current advancements in speech technology have posed a serious danger to the ASV system [49].

The ASVspoof2021 challenge [59], a follow-up to the ASVspoof2019 challenge, aims to create countermeasures to identify spoofed audio involving the coding and *transmission of text-to-speech* (TTS) [58], VC [39], and replayed attacks, with no released training or development data matching the telephony encoding and transmission artefacts encountered during evaluation. The most recent competition in this series, ASVSpoof 2021 [54], builds on the success of its predecessors. It has an extensive assessment framework that enables researchers to test their algorithms on a sizable dataset that represents spoofing assaults that occur in the real world. Therefore, the need for dependable speaker verification methods and spoofing countermeasures is critical [15]. The challenge covers a wide spectrum of possible risks to ASV systems by incorporating several spoofing techniques, including replay attacks, VC, and SS. ASVspoof 2021 is 4^{th} in a series of biannual, competitive challenges where the goal is to develop countermeasures capable of discriminating between *bonafide* (BNF) and *spoofed* (SPF) or DF speech [23]. ASVspoof 2021 comprises three major tasks.

LA: LA is used to describe the restricted and approved access to the ASVspoof dataset and related resources for the purpose of research and competition participation [87]. LA controls are set up to protect the security and integrity of the data while ensuring that only approved researchers or participants have access to the dataset and associated materials.

PA: PA is used to describe the restricted access to the hardware and physical infrastructure used in the competition [1]. Keeping the security and integrity of the physical resources involved entails making sure that only authorised individuals may physically access the ASVspoof 2021 systems [88], tools, and facilities. The physical area where spoofing assaults are recorded is then played again using varying-quality replay devices inside the same physical area.

DF: The DF intends to encourage research and development efforts to fend off spoofing assaults' rising level of sophistication [79]. In real-world contexts where DF technology is common. The assessment of anti-spoofing techniques against DF samples advances the *state-of-the-art* (SOTA) in spotting artificial or manipulated speech, making ASV systems more dependable and secure [4]. ASV systems are facing serious competition from DF technology, which is evolving rapidly. Malicious actors can produce SS samples that replicate the voice and speech features of a target person by using *deep neural networks* (DNN), *long short term memory* (LSTM), *generative adversarial networks* (GANs) and deep generative models [16].

The primary contributions of this endeavor include the following.

- The study investigated the use of hybrid combinations of the speech features of the ASVspoof databases using MFCCs, Mel-scaled spectrograms, chromagrams, spectral contrast, and Tonnetz in classifying people's speech recognition through speech utterances.

- Applying ML methods (k-NN, SVM, RF, GB, AdaBoost, XGBoost, MLP), the hybrid speech characteristics are concatenated and used to improve the classification accuracy of the ASV system.

- In this study, seven distinct ML algorithms are looked at. The study also sought to identify the number of MFCCs, chromagrams, Mel-scale spectrograms, spectral contrast, and Tonnetz, as well as the appropriate signal frame size and frameshift.

- This particular investigation focuses on voice recognition during spoken utterances and was carried out using ASVspoof 2021 LA. The issue is resolved using several ML algorithms that are based on balancing the data, and the majority of the techniques are enough to handle classification and regression issues.

- These ideas are experimental and are contrasted with cutting-edge methods. For all contexts and languages, the voice recognition results produced by ML algorithms demonstrate appreciable increases in ASV system performance compared to SOTA techniques. With over 99% accuracy, DL algorithms are more precise than ML algorithms.

Shortly, the study may include transformer-based models and reinforcement learning techniques to provide a more comprehensive comparison and help highlight the strengths and weaknesses of different approaches. Through the detailed analysis of experimental studies, including performance metrics, model interpretability, and robustness to hyperparameters, one can offer deeper insights into certain models that perform better than others under specific conditions. Transformers are tailored for image patches and rely on self-attention mechanisms to capture global relationships in an image's spatial structure. However, speech features are better represented using domain-specific models like CNNs, which excel at recognising local and hierarchical patterns. In this study, the traditional models performed well. The transformers and reinforcement learning models are computationally expensive and require more training data and longer training times due to their attention mechanisms and large number of parameters. Given the ASVspoof dataset, relatively minor scale, resource-intensive models would likely not justify the marginal improvement over the existing models. This would be especially true since our models DNN Single, DNN-CNN, DNN-BiLSTM, and DNN-ConvLSTM are already reaching high-performance levels.

The organisation of the paper is as follows: Section 2 provides a literature overview of ASVspoof. Section 3 introduces the database description. The feature extraction has been discussed in Section 4. Section 5 will explain the methodology of the speech recognition process, which includes the algorithm of ML and DL. In Section 6, the results and analysis have been discussed using the confusion matrix and graphs. Section 7 gives a brief summary of the paper, which is presented in the conclusion section, and future directions.

2 Literature Review

Today, a wide range of applications for *speaker recognition* (SR) employ voice-based technologies. In the future, there will be extensive research done on the topic of parodying and against mocking the ASV framework. The present development in the ASV system generates interest in securing these voice biometric-based systems for real-world uses [27]. The literature on spoofing detection, innovative acoustic feature representations, DL, end-to-end systems, etc., is included. Additionally, it summarises earlier research on spoofing assaults that put pressure on SS, VC, and replay, as well as current

initiatives to provide defences against spoof speech detection and speech sound disorder jobs. Currently, the majority of voice spoofing detection techniques use specialised algorithms that are solely concerned with LA or PA assaults. However, there is no previous knowledge of the kinds of spoofing assaults that really occur. As a result, academics begin to create generalised techniques to identify assaults, independent of the tactic employed to launch them [68]. Spectral-log filter-bank and relative phase shift characteristics were utilised as input to a model integrating a *deep neural network* (DNN) with an SVM classifier in several notable prior techniques against LAs [5]. Early studies on creating spoofing countermeasures were all performed using datasets that were specifically gathered and often produced using a small number of well-built spoofing attack methods [20]. Early research required practices since there were no widely used benchmark datasets. However, this brings up three issues [12]; first, repeatable research and meaningful comparisons of findings from many research teams can only be supported by the usage of shared datasets. Second, a spoofing assault can never be understood in advance in practice. Hence, a priori knowledge of a spoofing attack does not represent this reality [44].

Third, the most generalization-oriented defences may not be those developed utilising just a few spoofing assaults or spoofing algorithms [62]. By Wu *et al.* [87], the ASVspoof challenge series is introduced along with a description of the ASVspoof datasets, assessment procedures, and performance indicators [59]. The many spoofing attacks and the creation of defenses against them are covered. Liu *et al.* [53] provide an in-depth analysis of spoofing attacks and defenses in relation to ASV. Techniques covered include VC, SS, replay assaults, and VC with ML. In Tak *et al.* [78], the ASVspoof 2019 competition, with an emphasis on SPF and DF speech recognition, is presented. The creation of anti-spoofing systems, assessment methodologies, and dataset-gathering techniques is all covered, and it also discusses the effectiveness of various techniques and offers suggestions for new lines of investigation. By Gomez-Alanis *et al.* [24], the use of DL approaches in ASVspoof, especially for spoofing and anti-spoofing tasks, is examined. *Convolutional neural networks* (CNNs) [76] and *recurrent neural networks* (RNNs) [25], among other DL models, are discussed along with how well they function to identify spoof speech. The limitations and difficulties of DL-based anti-spoofing techniques are also examined in the research. Jung *et al.* [36] summarise the most current developments in spoofing and anti-spoofing strategies of an outline of the ASVspoof 2021 challenge. Feature representations, modelling tactics, and fusion techniques are only a few of the cutting-edge methods covered. The obstacles and upcoming developments in the sector are also highlighted in the study.

The main objective is to promote ASV, also known as speaker authentication or voice biometrics, by creating and assessing anti-spoofing technology.

- *Spoofing Attacks:* The primary goal of ASVspoof 2021 is to defend ASV systems against spoofing attacks. Attacks that use spoofing entail pretending to be a real speaker or tricking the system using fake or repeated speech recordings. The contestants create techniques to identify and categorise these spoofing assaults.
- *Dataset:* These datasets include real speech recordings from several speakers as well as spoofing attempts in a variety of formats, including TTS, VC, and replayed speech.
- *Evolution Metrics:* Multiple measures are used to assess the effectiveness of anti-spoofing systems. The min-DCF is the main metric employed in ASVspoof 2021 [24]. The *false alarm rate* (FAR) and the *missed detection rate* (MDR) are two factors this metric considers when assessing the system's performance.
- *Anti-Spoofing Techniques:* For the creation of successful anti-spoofing systems, participants in ASVspoof 2021 use a variety of methods. These approaches may include *feature extraction* (FE) algorithms, ML models (such as DNNs) [90], and fusion strategies for integrating several classifiers. To increase the resilience of their systems, participants frequently make use of developments in voice and audio processing, *pattern recognition* (PR), and ML.

LA: When discussing ASV, the term *"logical access"* (LA) refers to restricted and authorised access to the ASV system's software, databases, configuration settings, and other digital resources. In order to guarantee that only authorised parties or persons may interact with the ASV system and its related components, maintaining user accounts, permissions, and authentication procedures is necessary. LA restrictions are put in place to safeguard the ASV system's availability, confidentiality, and integrity as well as the sensitive data it processes. By ensuring that only authorised users may use the ASV system for authentication and verification purposes, these controls help prevent unauthorised access, data breaches, and other misuse of the system. A telephone banking service is an example of how an attacker may connect, bypass the microphone, and deliver converted or synthesised speech signals straight to the ASV system. In the communication channel post-sensor, this is referred to as audio insertion [83]. The LA of ASVspoof 2021 is focused on the creation of spoofing countermeasures that are resilient to transmission channel and codec variations [83]. The transmission of real and fake speech data generated by TTS, VC, or hybrid algorithms (VC systems fed with synthetic speech) over a *public switched telephone network* (PSTN) or a *voice over Internet protocol* (VoIP) network makes use of a particular codec and methods for automatically identifying speakers before being subjected to spoofing countermeasures. The major metric will be the t-DCF [19, 53].

PA: The term *"physical access"* (PA) refers to restricted and authorised access to the physical facilities, hardware, and infrastructure connected to the ASV system. It involves putting in place security measures to protect physical resources from unauthorised access, manipulation, theft, or damage. The security, integrity, and availability of the ASV system and its related components must be maintained through the use of PA restrictions. The danger of physical security breaches is decreased by these measures, which guarantee that only authorised individuals have PA to the tools, buildings, and data storage sites [59]. PA controls entail restricting access to specific locations that house the servers, storage, and hardware components of the ASV system. Access is prohibited to unauthorised persons, but authorised professionals, such as administrators, technicians, or security staff, are given access. To prevent unauthorised entrance and monitor activity within the building, these facilities are outfitted with security equipment, including alarm systems, access control systems, and surveillance cameras. PA restrictions frequently involve monitoring and surveillance equipment to track activity and identify any unauthorised entry attempts. Surveillance cameras, intrusion detection systems, and alarm systems are used to improve the overall physical security of the ASV system.

DF: The words "deep learning" and "fake" are combined to form the name *"Deepfake"* (DF). It sprang to popularity in the realm of visual media, when *Artificial Intelligence* (AI) algorithms were employed to produce doctored movies or photographs that show people talking or acting in ways they never actually did. The idea has been expanded to the audio realm, though, and now includes the production of artificial or altered speech using DL techniques. According to the codec and its setup, this procedure creates distortions. We plan to promote solutions for the identification of DFs in compressed audio used in television and media posted on news websites and social media platforms, among other generic end-user applications [53]. DF technology uses AI algorithms to produce incredibly realistic and convincing audio content that may be used to trick ASV systems or pass for real people [64]. Malicious actors can try to get around ASV systems, get unauthorised access, or pass themselves off as someone else by producing synthetic voice samples that sound like the target speaker. The inclusion of DF speech samples in ASV datasets or testing of ASV systems against DF attacks aims to advance the field's comprehension of DF technology, its implications for ASV, and the creation of efficient countermeasures to improve the security and dependability of ASV systems. In order to advance the understanding of DF technology, its implications for ASV, and the development of effective countermeasures to enhance the security and dependability of ASV systems in ASV datasets or test ASV systems against DF attacks [64].

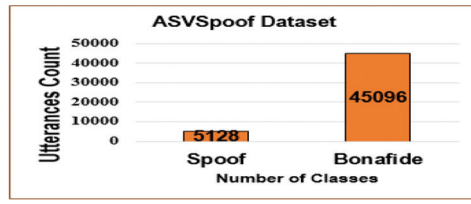

Figure 1: Block diagram of ASVspoof Dataset of the robust Spoof and Bonafide detection.

3 Database Description

The ASVspoof 2019 challenge,[1] which comes with a fresh dataset, makes a number of improvements over earlier iterations. It is the first to take into account SS, VC, and replay as separate spoofing attack types. The ASVspoof 2019 collection of real and fake speech signals includes SS and converted voice signals produced using the most recent, cutting-edge technology. The finest of these algorithms can provide voice spoofing that is perceptually identical to real speech when used in carefully controlled environments. Thus, the goal of ASVspoof 2019 is to establish if improvements in SS and VC technologies constitute a bigger danger to the dependability of ASV systems or whether, alternatively, they can be reliably recognised with current countermeasures. The database can be observed from Figure 1.

ASVSpoof2021 database consists of three new evolution partitions for LA, PA, and DF tasks.[2] Compared to earlier iterations, ASVspoof 2021 is purposefully more challenging. The DF task is new to ASVspoof and expands the initiative's objectives to include the detection of spoof speech in situations outside of ASV. The DF task simulates a situation in which an attacker has access to the speech data of a target victim, such as information shared on social media. A renowned person, a social media influencer, or a simple individual might all be the victim. This article offers a description of the challenge findings, the four baselines for the challenge, the three objectives, the new databases for each of them, the assessment metrics, and the evaluation platform. Even though the complexity has increased due to the inclusion of channel and compression variations, the results for the LA and DF tasks are comparable to those from past ASVspoof editions. Using public data and speech DF technology, the attacker is anticipated to produce spoof speech that mimics the victim's voice. The recordings will then be posted on social media, in call centres, or in any other application that supports this sort of behaviour.

ASVspoof, created as part of the 2015 Interspeech anti-spoofing challenge, contains only synthetically generated spoofing attacks [42]. These attacks are

[1] https://www.asvspoof.org/index2019.html.
[2] https://www.asvspoof.org/index2021.html.

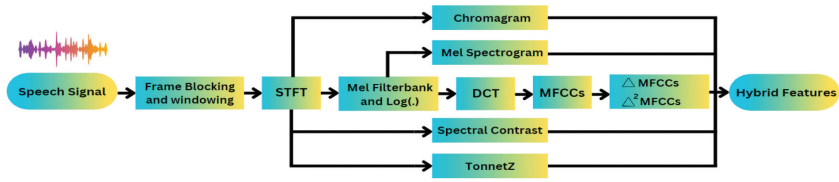

Figure 2: Block diagram for Hybrid combination of the feature extraction process.

assumed to be fed into a verification system directly bypassing its microphone and are also coined as LA attacks [86]. It is important to note that all data utilized for training and evaluation were simulated using acoustic replay simulation in accordance with the evaluation strategy [46]. The RedDots database [48] serves as the primary source of real recordings for the ASVspoof 2017 challenge, and the RedDots Replayed database [41] serves as the primary source of spoof replay recordings. Part 01 of the original corpus, consisting of 10 typical short sentences, was replayed across various recording settings and recording equipment to form the RedDots Replayed corpus [22]. The performance of replay attack detection is found to be significantly enhanced by feature normalization. As a result, we employ spectrum analysis based on DFT in our study [3]. The challenge consists of two conditions: a common condition in which only ASVspoof 2017 data may be used to train detection systems, and a flexible condition in which any external data may be utilized. ASVspoof 2019 contains evaluation measures such as the $t - DCF$ and EER. These parameters are employed to rate how effectively ASV systems find spoofing attempts. Additionally, benchmarks and contests were a part of the ASVspoof 2019 Challenge to promote the advancement of powerful spoofing detection methods [82]. As LA and PA attacks, respectively, the ASVSpoof 2019 database includes both synthetic and replay speech assaults. These two tracks have three subsets, namely the train, development, and evaluation sets [17].

4 Feature Extraction

The speakers' vocal tracts' structural variations constitute a biometric identification trait and an intrinsic property. By taking *feature vectors* (FVs) from the training utterances and using them to create reference models, the training process allows the system to become familiar with the speech characteristics of the registered speakers. Similarly, FVs are extracted from the test utterance during testing, and the degree of similarity between them and the reference is assessed using a matching algorithm. The ASVspoof pipeline starts with preprocessing the audio data. To do this, we first use a pre-emphasis filter

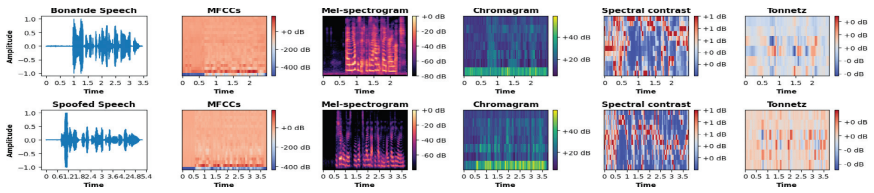

Figure 3: Illustrating the difference between BNF and SPF speech utterances. The extracted speech features are summarized by using the MFCCs, Mel-spectrograms, Chromagrams, Spectral contrast, and Tonnetz features.

to remove any low-frequency noise from the audio stream before dividing the signal into frames of the same size. The typical frame interval is between 20 and 30 milliseconds, with a 50% gap between each frame. Then, MFCCs are extracted based on the finding that human speech is represented as a time-varying linear filter. A bank of triangle filters with logarithmically separated frequency bands is then applied to extract the MFCC characteristics after computing the *short-time Fourier transform* (STFT) of the spoken stream. The output of the filter bank is then transformed using the *discrete cosine transform* (DCT), and a subset of the resultant coefficients is retained as features, shown in Figure 2.

MFCCs Mel-frequency for FE and voice recognition. This is accomplished by first applying an STFT to the signal, which yields an audio spectrum. The spectrum is then converted into the Mel-scale using a bank of Mel scale filters [50]. The MFCCs are then produced by taking the spectrum's logarithm and applying a DCT. Each lengthy speech frame is subjected to the DCT, and the DCT coefficients are then organized into subbands. Each subband then undergoes a linear prediction analysis in the frequency domain [84]. The provided spoken utterance is first broken up into tiny speech frames with a frame size of 20 msec and a frameshift of 10 msec in order to extract MFCCs. Based on the application of the DCT to the log power spectrum on a nonlinear Mel scale of frequency, MFCC coefficients indicate the short-term power spectrum of the speech signal [61]. The windowing technique is used to lessen the gap between speech frames at their beginning and conclusion. The time domain signal is then converted to the frequency domain by applying the STFT to each frame. The Mel-scale filter banks are used in Eq. 2 to compute all of the frequencies derived from the FFT.

$$y(k) = \sum_{s=0}^{S-1} y(s) e^{\frac{-j2\pi ks}{S}}. \tag{1}$$

The energy at each Mel-frequency value is then computed as a logarithm, and all of the log-Mel-spectrums are then converted back to time using the DCT. The Fourier-based features also offer excellent temporal frequency analysis

potential if we concurrently increase the number of frames and the number of bins per frame [51]. The resultant spectrums' derived amplitudes are referred to as MFCCs.

$$F_{Mel} = 2595log(1 + \frac{f}{700}). \tag{2}$$

Where f represents the frequency. Since there are 320 samples in each speech frame, 512 frequency bins were selected as the number of DFT coefficients. We obtained the compressed, orthogonalized energy vectors of the Mel-filter bank as a feature vector by selecting the first 13 coefficients from DCT, removing the 0^{th} coefficient, and then applying their DCT.

$$p(n) = \sum_{m=0}^{M-1} F_{Mel}(m)cos(\frac{\pi n(m - \frac{1}{2})}{M});\ n = 0, 1, \ldots, P - 1, \tag{3}$$

where, $p(n)$ are the cepstral coefficients and P represents the number of coefficients. i.e., $p_t(i)$

$$\Delta p_t(i) = \frac{\sum_{m=-r}^{r} m.p_t(i)}{\sum_{m=-r}^{r} m^2}, \tag{4}$$

$$\Delta^2 p_t(i) = \frac{\sum_{m=-r}^{r} m.p_t(i)}{\sum_{m=-r}^{r} m^2}. \tag{5}$$

The 1^{st} order derivative $\Delta p_t(i)$ and the 2^{nd} order derivative $\Delta^2 p_t(i)$ features are generated after extracting 13 dimensions from the $p_t(i)$ coefficients and they are then stacked along with 0^{th} an average energy for each speech frame coefficient to create a $[0^{th}, p_t(i), \Delta p_t(i)$ & $\Delta^2 p_t(i)]$, a 40-dimensional feature vector obtained. The feature data shows higher timing variation features.

Mel-spectrogram When the frequencies are converted into the Mel scale, the spectrogram is called a Mel-spectrogram. The given speech utterance is divided into frames with a frame size of 20 msec and an overlap of 10 msec to extract the Mel-spectrogram's features. Each speech frame was then given a window, and each one had an FFT applied to it. The frequency spectrum is divided into equal space frequencies to create the Mel scale for the speech utterance and to obtain the Mel-spectrogram; the data was then passed through filter banks. Indicate the input signal using $f\epsilon\ \theta^R$, window functionality is generated $g\epsilon\ \theta^R$ and the Mel filters, typically given by triangular functions by $\Lambda_\nu \epsilon\ \theta^R\ \forall\ \nu\ \epsilon I, I = 1, 2, \ldots, K$, where K is the selected number of Filters. Hence, the Mel-spectrogram is given by

$$MS_g(f)(b, \nu) = \sum_k |F(f.T_b)|^2.\Lambda_\nu(k). \tag{6}$$

Chromagram A chromagram displays the changing pitch of voice input over time. Any spectrogram or STFT can be utilized as input in place of FFT.

Harmony and pitch classes are represented differently using chromagram characteristics [7]. The binning approach used to extract 12 distinct pitch classes from the supplied voice utterance in order to retain chroma characteristics and a unique steganalysis model based on *spatial and temporal feature fusion* (STFF) for each speech frame [21]. It is applied to encrypt harmony while reducing loudness, octave height, or timbre fluctuations. When constructing a speaker's voice print for the purpose of classifying speech emotions, a popular hand-designed approach mostly relies on folding several octaves of a spectral representation into a 12-semitone chromagram. A total of 12 semitone chromagrams were taken into account. To capture different facts of the audio, several statistical or perceptual characteristics can be calculated using the chroma representation. Mean, variance, energy, and harmonic-related characteristics like pitch histograms or harmonic pitch class profiles are often used features.

$$HPCP_b = \sum_{m=1}^{M} CQT[b + 12m], \tag{7}$$

where, $1 \leqslant b \leq 12$ and M number of octaves involved.

Spectral Contrast In audio signal processing, the perceptual difference between various frequency bands in a sound source is captured using the spectral contrast FE approach. When performing tasks like voice recognition, music genre categorization, and sound event detection, it offers details about the relative energy disparities across various frequency ranges. These characteristics are determined by computing the root mean square difference between the speech frames' spectral evidence and spectral peak. Neighborhood criteria and octave scale filters are used to quantify spectral contrast in subbands. In general, narrow-band signals with high contrast values are clearer than noise with low contrast values. The mean, standard deviation, and spectral peak of all frames are employed as the spectral contrast characteristics to describe the full piece of music [67]. The strength of the spectral peaks P_k and spectral valleys V_k are estimated as

$$P_k = log(\frac{1}{\alpha N} \sum_{i=1}^{\alpha N} x_{k,i}), \tag{8}$$

$$V_k = log(\frac{1}{\alpha N} \sum_{i=1}^{\alpha N} x_{k,N-i+1}). \tag{9}$$

And the difference is

$$SC_k = P_k - V_k. \tag{10}$$

Tonnetz The Tonnetz is a pitch based on melodic contribution; tight symphonic connections are displayed as tiny separations. This is the six-dimensional pitch space that depicts the resonant pitch interactions in falling

and increasing voice signals, too. Each speech frame's pitch characteristics in a spoken utterance play a crucial role in identifying the speech's emotions. Tonnetz characteristics may be used as data for a number of ML techniques. Tasks like music analysis, composition, or recommendation are made possible by these models' ability to be trained to spot patterns or extract useful information from Tonnetz representations. By combining some of the chroma feature classes, which totalled 12 classes, Tonnetz, which is also known as tonal centroid features, divides pitch into 6 separate classes. As a result, it is computed using a 6-dimensional basis and chroma characteristics that were projected onto a chromagram. The mathematical expression for Tonnetz is given below

$$\psi_t(d) = \frac{1}{||C_t||} \sum_{l=0}^{11} S(d,l) C_t(l), \quad 0 \leq d \leq 5, \tag{11}$$

$$S_l = \begin{bmatrix} a_1 sin(\frac{7\pi l}{6}) \\ a_1 cos(\frac{7\pi l}{6}) \\ a_2 sin(\frac{3\pi l}{2}) \\ a_2 cos(\frac{3\pi l}{2}) \\ a_3 sin(\frac{2\pi l}{3}) \\ a_3 cos(\frac{2\pi l}{3}) \end{bmatrix}, \quad o \leqslant l \leqslant 11. \tag{12}$$

The amplitude of the signal varies over time in BNF speech, but it is less variable in SPF speech. In comparison to SPF MFCCs, BNF MFCCs show fewer fading graphs, and the SPF speech's Mel-spectrogram is more dispersed. When compared to BNF, SPF speech has a higher chromagram scatter value with respect to time. In contrast to genuine speech, spoofed speech has a dense value of spectral contrast. According to Tonnetz, genuine speech has far lower wavelength visibility than faked speech. The speech features are focused on Mel-spectrograms, Tonnetz, Chromagrams, and MFCCs and are pictorially represented in Figure 3. Provides insights into the values of each feature in the given speech utterance.

5 Methodology

A given spoken utterance can be used to best categorise the speech. The offered spoken utterances are both authentic and fake, categorised using the k-NN, SVM, RF, *Gradient Boosting* (GB), AdaBoost, *Extreme Gradient Boost* (XGBoost), and *Multi-Layer Perceptron* (MLP) classification models. In this paper, our aim is to detect DF audio using the ML algorithms mentioned above. Decisions made by ML models are based on associations found in training data. Models can pick up unrelated inputs, artefacts, or confounders during

training. Such artifacts often help to achieve good results by overestimating real performance in a test set, unless they are explicitly taken into account during training and inference [14]. The quality and variety of training data, feature engineering, algorithm optimization, and the prevalence of certain types of spoof attacks in the dataset are all aspects that affect how successful these algorithms are. The objective is to classify fake and real speech through ML and DL methodology.

***k*-NN:** The k-NN method determines the K=3 closest data points to a test point, which then assigns the majority of test points to a class. In order to calculate how similar two data points are to one another, *Euclidean distance* is used, which is defined as:

$$d(x_i, x_j) = \left(\sum_{p=1}^{n} w_p (a_p(x_i) - (a_p(x_j)))^2 \right)^{\frac{1}{2}}. \tag{13}$$

Where a vector $x = (a_1, a_2, a_3, \ldots, a_n)$, n is the dimensionality of the vector input, namely the number of sample's attribute, a_p is the sample's p^{th} attribute, w_p is the weight of the p^{th} attribute, p is from 1 to n, the smaller $d(x_i, x_j)$ the two samples are more similar.

The test sample's class label is chosen by its k closest neighbors with a majority vote.

$$x(d_i) = argmax_k \sum_{x \epsilon kNN} y(x_j, c_k) \tag{14}$$

where d_i denotes a test sample, x_j denotes one of its k closest neighbors in the training set, and $y(x_j, c_k)$ denotes whether x_j belongs to $class_{c_k}$.

SVM: To distinguish real speech from speech that has been replayed, a binary SVM is trained using vector characteristics [2]. In higher-dimensional space, SVM divides data into two groups as efficiently as possible using the hyperplane. The margin is the difference between the hyperplane and the closest data from each class [74]. The equation of a hyperplane in a d-dimensional space is defined by

$$h(x) = sign(w^T x + b). \tag{15}$$

Where w signifies the weight vector (orthogonal to the hyperplane), and b denotes a bias term.

The distance between a point x and the hyperplane can be computed as:

$$d(x, h(x))) = \frac{w^T x + b}{\|w\|}, \tag{16}$$

where $\|w\|$ denotes the Euclidean norm of the weight vector w.

To ascertain the hyperplane with the maximum margin, it is necessary to solve the optimization problem:

$$d_{min} = \frac{\|w\|^2}{2}, \tag{17}$$

with constraints, $y_i(w^T x_i + b) \geq 1 \forall\ i = (1, 2, 3, \ldots, n)$ After solving the above equation, the maximum margin problem is reduced to $Margin = \frac{2}{\|w\|}$. The tunable hyperparameters considered for classification tasks are $C = 0.1$, gamma=0.001, and 'kernel'=*poly*.

Random Forest (RF) is an ensemble learning method that enhances prediction accuracy and reduces variance by aggregating the outputs of multiple decision trees (DTs) trained on random subsets of data and features. To minimize overfitting, each tree is constructed using bootstrap sampling and random feature selection. The final prediction is obtained via majority voting across all trees. In this study, the model employs 201 estimators with entropy as the splitting criterion.

RF: It is a potent method for solving problems that enhances prediction accuracy and reduces variance by aggregating the outputs of multiple *decision trees* (DTs) trained on random subsets of data and features. To optimize overfitting, each tree is trained on different subsets of data and with random feature selection. The final prediction is obtained using a majority vote system across all DTs, and the model employs 201 estimators with entropy as the splitting criterion.

Assume that k is the node, X_j is the significance of the characteristics, and Y_k is the total samples for all nodes. The significance of an equation can be expressed as [26]:

$$X_j = \sum k : jY_k G_k. \tag{18}$$

Where nodes k divide on feature j. By normalizing first the X_j for each tree in the RF and then adding those normalized values for each tree, the final significance of the feature X_j for each feature is determined.

GB: Build predictive models by combining weak learners, such as DTs, into *gradient-boosted trees* (GBTs) whenever a DT acts as a poor learner. The progressive optimization technique *Gradient descent* (GD) determines the minimum of a function. The purpose of GD is to identify the set of input variables that minimize a cost function, sometimes referred to as an objective function or loss function [65].

GD optimizes model parameters by following the direction of the steepest descent, determined by the gradient of the cost function. The step size of each update is controlled by the learning rate (set to 0.1), where inappropriate values can lead to slows convergence or overshooting. With weak and strong models, the widely used GB can solve the issues by reducing the loss function,

Table 1: Comparative analysis of diverse ML models utilizing time and space complexity, training time, and hyperparameter configurations on the SAA and AccentDB databases, respectively.

Model	**Hyperparameter**	**Time Complexity**	**Space Complexity**
k-NN	n_neighbors = 3	O(l) (trn), O(n) (tst)	O(n)
SVM	C=0.1, gamma=0.001, kernel=poly	O(n^3) (trn), O(T) (tst)	O(n^2)
RF	n_estimators=201, criterion=entropy	O($T \times nlogn$) (trn), O(T) (tst)	O($T \times n$)
MLP	random_state=1, max_iter=300	O($l \times n$) (trn), O(l) (tst)	O($n + h^2$)
GB	learning_rate=0.1, n_estimators=100 max_depth =3	O($T \times n$) (trn), O(T) (tst)	O(T)
AdaBoost	n_estimators=50, learning_rate=1.0	O($T \times n$) (trn), O(T) (tst)	O(T)
XGBoost	n_estimators=100, learning_rate=1.0 max_depth=4, alpha=10	O($T \times n$) (trn), O(T) (tst)	O(T)

often referred to as the sum of the residuals from the weak models. The *mean squared error* (MSE), is the most often used loss function. MSE is defined as:

$$M(r, f(s)) = (r - f(s))^2, \tag{19}$$

where r is the value of the target, and $f(s)$ is the predicted value of the target variable by the model.

The most commonly used loss function is the log loss function, which is defined as:

$$M(r, f(s)) = \frac{-1}{N} \sum_{i=1}^{K} \sum_{j=1}^{L} r_{ij} * log(s_{ij}). \tag{20}$$

The GB approach adds weak models to the ensemble iteratively, with each weak model being trained to reduce the residual of the preceding models. The number of estimators is 100, and the maximum depth used for this model is 3.

AdaBoost: In most cases, the DT is used as the basic model; however, additional models include SVM and *logistic regression* [56]. Each time a boosting iteration is performed, the training data points are modified by applying weights $\{w_1, w_2, w_3 \ldots, w_N\}$. In the first, scaled weights are used, $w_1 = \frac{1}{N}$. The weights of the data points are changed for each subsequent iteration depending on whether the data is correctly labeled or not. Models are produced one after the other, and the success of the earlier models has an impact on the development of the later ones. The ultimate forecast is then created by weighing the majority's votes on all of their guesses together. For the base learner, it is set as $m := m + 1$ and then computes the base learner with a weighted data set [55].

Now, the new weight is observed as:

$$w_i^{[m-1]}, \ldots, w_N^{[m-1]} \rightarrow \hat{h}^{[m-1]}(\cdot). \tag{21}$$

Deduce the error rate and update the iteration's particular coefficient $\alpha_m \rightarrow highvalues$ due to a low error rate. Modify the individual weights and $w_i^{[m-1]} \rightarrow highervalues$ whenever an observation is mislabeled. Hence, the new observation is x_{new}:

$$\hat{f}_{AdaBoost}(x_{new}) = sign(\sum_{m=1}^{m_{stop}} \alpha_m \hat{h}^{[m]} x_{new}). \tag{22}$$

Here, m_{stop}, after iterating the steps of Eq. 21 and the above statement. The number of estimators is 50, and the learning rate of 1.0 is used for this model.

XGBoost: XGBoost outperforms the GB framework by generating trees concurrently rather than sequentially. Its main goal is to prevent overfitting and facilitate accurate predictions by minimizing the sum of a loss function and a regularization term. To create a robust and effective model, XGBoost

combines DTs with GB methods. The number of estimators is 100, the learning rate is 1.0, the alpha is 10, and the maximum depth is 4, which are used for this model. By optimizing systems and enhancing algorithms, XGBoost outperforms the GB framework by generating trees concurrently rather than sequentially. Its main goal is to prevent overfitting and facilitate accurate predictions by minimizing the sum of a loss function and a regularization term. To create a robust and effective model, XGBoost combines DTs with GB methods.

MLP: The input layer's neurons represent the number of input features, and the output layer's neurons correspond to the number of classes. Optimizing the network architecture, including layers, neurons, and connections, is key to achieving effective classification. In this random state, it is taken as 1, and the maximum iteration is 300. In summary, the trade-offs between these models, considering factors like computational complexity, space complexity, number of estimators, maximum depth, random state, gamma, and kernel, etc., can be observed from Table 1.

Motivation to use DL Algorithm: When used to thoroughly investigate and methodically study sets of characteristics obtained using well-established signal processing techniques and algorithms, ML models have demonstrated exceptional performance [28, 57]. CNNs, RNNs, and transformer-based models are examples of these developments that offer increased capabilities for spotting DF audio [60]. However, in the present research environment, FE methods based on DL are mostly responsible for the most cutting-edge developments in a variety of prediction and learning applications, including audio, video, and text data.

DL involves the technique of matrix manually multiplying created dense FVs into larger FVs [82, 83]. The output of the DL model is a dense FV that is produced by the last hidden layer and is standardized [35]. Certain DL algorithms specifically construct their last layer for classification or regression tasks by integrating the proper transformation function [8, 47, 73]. A DL model's architecture may be altered to fit the needs of the particular application, offering flexibility in terms of the number of layers, dimensions of hidden layers, utilized transformation functions, and selected loss function [6, 43]. To ensure computational and temporal efficiency, we created custom DL architectures for our research that were suited to our dataset.

DNN-Single: DNN Single architecture [40] is used to investigate and comprehend the performance of extracted FVs. The first hidden layer of architecture encounters the FV of dimension 193, which converts it into a vector of dimension 512 using weighted matrix multiplication and the standard ReLU transformation function. With the same transformation function, additional hidden layers with output dimensions of 256 and 128 are also used. The dropout transformation is applied between each hidden layer. Dropout is a common technique for dealing with extreme variance that leads to overfitting

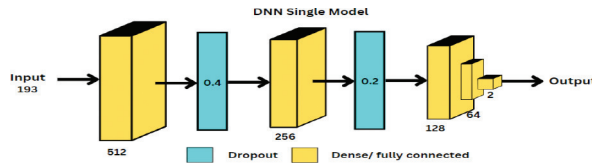

Figure 4: Block architecture of DNN-single Deep Learning Model.

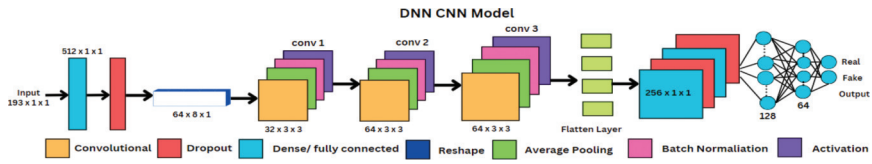

Figure 5: Block architecture of DNN CNN Deep Learning Model.

of DL models. To put it into action, it simply removes a fractional ratio of weight from the hidden layer, preventing over-training in all layers. As a result, Dropout with a ratio of 0.4 is used in between. Using a hidden layer, the 128-dimensional vector is transformed into a 64-length vector. As it is a multi-class classification, the final 64-dimension dense vector is used for two-class classification using the softmax classification technique noticed from Figure 4. The cross-entropy loss function is used in the training. In loss minimization, the Adam optimization technique is used to reduce the loss [71]. In the CNN training procedure, we were interested in the frequency area that is more useful for differentiating authentic and repeat speech [45].

The model is designed for classification tasks with two output classes., where the input layer accepts a feature vector $s \in \Re^{193}$, representing the dataset's attributes. The *dense layer* (dL) 1^{st} maps the input to a 512-dimensional feature space using the transformation $\pounds_1 = \mu_1 s + \alpha_1$, where $\mu_1 \in \Re^{512\times193}$ and $\alpha_1 \in \Re^{512}$ are training parameters initialized with the Glorot uniform initializer as $\theta \sim U\left[-\frac{\sqrt{6}}{\sqrt{n_{in}+n_{out}}}, \frac{\sqrt{6}}{\sqrt{n_{in}+n_{out}}}\right]$. The output $\xi_1 = ReLU(\pounds_1)$, where $ReLU(\pounds) = \max(0, \pounds)$, passes through a dropout layer with a rate of 0.4, which deactivates 40% of the neurons during training by applying $\pounds_1^{dropout} = \pounds_1 \odot m_1$, where $m_1 \sim Bernouli(p = 0.6)$. The dL 2^{nd} reduces the feature space to 256 dimensions using $\pounds_2 = \mu_2 \pounds_1^{dropout} + \alpha_2$, where $\mu_2 \in \Re^{256\times512}$ and $\alpha_2 \in \Re^{256}$, followed by $\xi_2 = ReLU(\pounds_2)$. A 2^{nd} dropout layer with a rate of 0.2 applies regularization. The dL 3^{rd} compresses the feature space further to 128 dimensions with $\pounds_3 = \mu_3 \pounds_2^{dropout} + \alpha_3$, where $\mu_3 \in \Re^{128\times256}$ and $\alpha_3 \in \Re^{128}$, followed by $\xi_3 = ReLU(\pounds_3)$. Correspondingly, the 4^{th} dL reduces the dimensions to 64 using $\pounds_4 = \mu_4 \pounds_3^{dropout} + \alpha_4$, where $\mu_4 \in \Re^{64\times128}$ and $\alpha_4 \in \Re^{64}$ with activation $\xi_4 = ReLU(\pounds_4)$. The final output

layer computes $\pounds_5 = \mu_5 \pounds_4 + \alpha_5$, where $\mu_5 \in \Re^{2\times 64}$ and $\alpha_5 \in \Re^2$. The softmax AF as $\hat{y} = softmax(\pounds_5)$, where $softmax(\pounds_i) = \frac{exp(\pounds_i)}{\sum_{j=1}^{J} exp(\pounds_j)}$, converts the output into a probability distribution over the two classes.

DNN-CNN: The DNN-CNN [89] model was developed to investigate and grasp the functionality of CNN-based architectures [9] in this particular field utilizing our generated features. We developed a DNN for replay noise classification and spoofing detection since replay noise is important [72]. The extraction of features is made possible by these convolutional procedures, which are then subjected to various pooling approaches and *batch normalization* (BN) algorithms to improve efficiency and prevent overfitting during training. Also connected to transformation functions are these extracted characteristics [93]. The initial and end hidden layer settings, dropouts, and transformations used in our combined DNN-CNN architecture are identical to those used in our DNN single model [75]. A three-dimensional matrix with the dimensions $(64, 8, 1)$ is created using feature vectors of dimension 512 to mimic image data. The first step is an average pooling with the same strides, followed by a convolutional hidden layer with 32 kernels of size 3×3 applied with a 2×2 stride. Then, two separate CNN layers with a 64 kernel size are applied. The output is transformed mathematically using a ReLu-based algorithm [18, 92] and BN after each CNN layer can be observed from the Figure 5. The result is then reduced to a two-dimensional vector, which serves as the input for the remaining layers, which are the same as those used in the DNN Single model.

The DNN-CNN, designed for binary classification, begins with an input layer accepting a feature vector $s \in \Re^{193}$ and provides input attributes. The 1^{st} dL computes $\pounds_1 = \mu_1 s + \alpha_1$, where $\mu_1 \in \Re^{512\times 193}$ and $\alpha_1 \in \Re^{512}$ are trainable parameters initialized with the Gorot uniform initializer. The output $\xi_1 = ReLU(\pounds_1)$, where $ReLU(\pounds) = \max(0, \pounds)$, is regularized by a dropout layer with a rate of 0.4, which deactivates 40% of neurons during training: $\pounds_1^{dropout} = \pounds_1 \odot m_1$, where $m_1 \sim Bernouli(p = 0.6)$. The resulting 512-dimensional output is reshaped into a $3D$ tensor of shape $(64, 8, 1)$ as input to the CNN. The CNN begins with a convolutional layer that applies 32 filters, each of size 3×3, with a stride of 2×2, computing feature maps $\pounds_2 = \mu_2 \star \pounds_1^{dropout} + \alpha_2$, where $\star$ denotes the convolution operation, $\mu_2 \in \Re^{3\times 3\times 1\times 32}$ and $\alpha_2 \in \Re^{32}$. The output is passed through ReLU activation and an average pooling layer with a pool size of 2×2 and stride 2×2, reducing the spatial dimensions by half. BN stabilizes the activations, producing $\xi_2 = ReLU(BN(\pounds_2))$. A 2^{nd} CL uses 64 filters of size 3×3 with '*same*' padding, preserving spatial dimensions, and reckons $\pounds_3 = \mu_3 \star \pounds_2 + \alpha_3$, followed by BN and ReLU activation: $\xi_3 = ReLU(BN(\pounds_3))$. This is repeated for another CL with 64 filters, resulting in refined feature maps ξ_4. The output is then flattened into a vector $\xi_5 \in \Re^n$, where n is the total number of features. The flattened vector passes through a dL with 256 neurons,

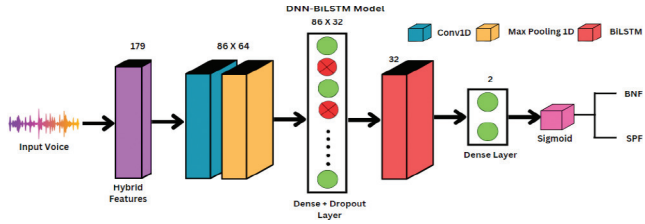

Figure 6: Block architecture of DNN - BiLSTM Deep Learning Model.

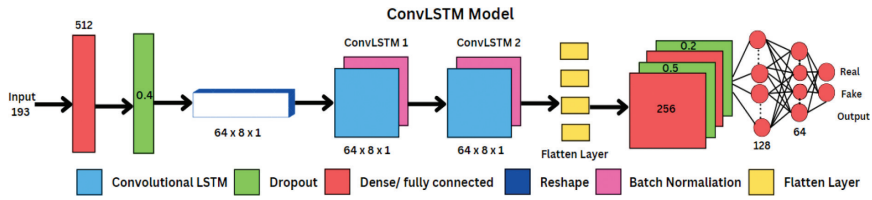

Figure 7: Block architecture of DNN-ConvLSTM Deep Learning Model.

computed as $\pounds_4 = \mu_4 \pounds_5 + \alpha_4$, where $\mu_4 \in \Re^{256 \times n}$ and $\alpha_4 \in \Re^{256}$, followed by $\pounds_6 = ReLU(\pounds_4)$. A dropout layer with a rate of 0.5 regularizes the output. The subsequent DN layers refine the feature space with $256, 128$, and 64 neurons sequentially, each applying $\pounds_i = \mu_i \xi_{i-1} + \alpha_i$ and $\xi_i = ReLU(\pounds_i$. The final dL outputs as $\pounds_{final} = \mu_{final} \xi_{last} + \alpha_{final}$, where $\mu_{final} \in \Re^{2 \times 64}$ and $\alpha_{final} \in \Re^2$, followed by a softmax activation: $\hat{y} = softmax(\pounds_{final})$, where softmax $softmax(\pounds_i) = \frac{exp(\pounds_i)}{\sum_{j=1}^{J} exp(\pounds_j)}$. The model combines dLs for feature extraction with CNN layers for spatial feature learning, effectively capturing spatial patterns for accurate classification.

DNN-BiLSTM Model: The DNN-BiLSTM model is designed to leverage the power of BiLSTM [13] layers in combination with dLs for sequence modeling tasks [32]. This architecture is particularly suitable for capturing temporal dependencies and patterns within sequential data. The model begins with an input layer, which takes in sequences of length 193. This input is then passed to a dL with 512 units and a ReLU *activation function* (AF), initialized with the Glorot uniform initializer [38].

A dropout [63] layer with a rate of 0.4 is applied to prevent overfitting. The resulting output is reshaped into a three-dimensional tensor of dimensions $(64, 8)$, mimicking the structure of a 2D image. BN [69] is then applied to normalize the tensor, followed by two BiLSTM layers with 64 units each. The BiLSTM layers process the input sequences in both forward and backward directions, allowing the model to capture dependencies from both past and future contexts. The first LSTM layer returns sequences, while the second LSTM layer returns only the final hidden state. The output from the BiLSTM

layers is passed through a dL with 256 units and a ReLU AF. A dropout layer with a rate of 0.2 is applied to further regularize the model. The subsequent hidden layers consist of two dLs with 128 and 64 units, respectively, using ReLU activation. Finally, the output layer with 2 units and a softmax [31] AF is added to classify the input sequences into two classes. The DNN-BiLSTM model shares the same set of performance indicators, including accuracy, precision, recall, and AUC-ROC score, as the DNN Single model. It utilizes the cross-entropy loss function [80] and the Adam optimizer for training, and softmax-based classification for multi-class classification tasks can be seen from the Figure 6.

The model combines dLs with BiLSTMs to process sequential patterns and extracts robust feature vectors $s \in \Re^{193}$ as input data with 512 neurons maps with high dimensional representation computed as $\pounds_1 = \mu_1 s + \alpha_1$, where $\mu_1 \in \Re^{512\times193}$ and $\alpha_1 \in \Re^{512}$ are trainable parameters initialized using Glorot uniform initialization. ReLU activation $\xi_1 = ReLU(\pounds_1)$, where $ReLU(\pounds) = \max(0, \pounds)$, is applied to introduce non-linearilty. To reduce overfitting, a dropout layer with a rate of 0.4 is applied, producing $\xi_1^{dropout} = \xi_1 \odot m - 1$, where $(m-1) \sim Bernouli(p = 0.6)$. The output is reshaped into a $3D$ tensor of shape $(64, 8)$, preparing it for sequential processing by LSTM layers. A BN layer is applied to the reshaped tensor, stabilizing activations and accelerating convergence. Then fed into the 1^{st} BiLSTM layer, which consists of 64 units in each direction. For each time step t, the forward LSTM determines $\overrightarrow{h_t} = LSTM(\overrightarrow{h_{t-1}}, s_t)$, while backward LSTM assess $\overleftarrow{h_t} = LSTM(\overleftarrow{h_{t+1}}, s_t)$. The outputs are concatenated to form $(h_t = \left[\overrightarrow{h_t}; \overleftarrow{h_t}\right])$ captures temporal dependencies in both directions. The 2^{nd} BiLSTM layer with 64 units further processes these outputs, resulting in a refined temporal characterization. The sequence output is concatenated into a single vector $h(LSTM)$, extracted from the final timestep outputs of the forward and backward passes, establishing a comprehensive characterization of the sequence.

The output obtained from the BiLSTM is passes through a dL with 256 neurons, determined as $\pounds_2 = \mu_2 \xi_{LSTM} + \alpha_2$, where $\mu_2 \in \Re^{256\times m}$ with m being the dimensionality of ξ_{LSTM} and $\alpha_2 \in \Re^{256}$. ReLU activation is applied by a dropout layer with a rate of 0.2 for regularization. The processed characterization is further refined through $2D$ dLs with 128 and 64 neurons, respectively, each applying ReLU activation: $\pounds_i = \mu_i \xi_{i-1} + \alpha_i$, where μ_i and α_i are the parameters of each layer. The final dL consists of 2 neurons and evaluates $\pounds_{final} = \mu_{final} \xi_{last} + \alpha_{final}$, where $\mu_{final} \in \Re^{2\times64}$ and $\alpha_{final} \in \Re^2$. A softmax activation produces probabilities for the two target classes. The combined dLs for feature extraction with BiLSTMs for temporal dependency modeling make it highly suitable for tasks involving structured or sequential data for classification performance.

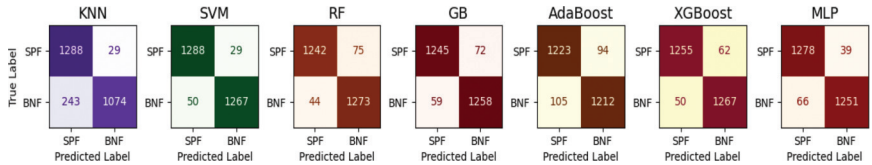

Figure 8: The confusion matrix shows the proportion of SPF and BNF predictions made using one of the spatiotemporal machine learning techniques using (a) KNN (b) SVM (c) RF (d) GB (e) AdaBoost (f) XGBoost (g) MLP.

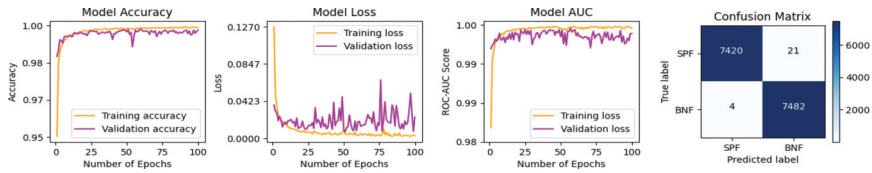

Figure 9: The graphical representation displays the accuracy, loss, and ROC-AUC score with a confusion matrix applied to the DL-based DNN Single model to the dataset.

DNN-ConvLSTM Model: Still, after achieving great results when using sequential deep learning architectures to this dataset, such as LSTM [29] or RNN (Recurrent Neural Network) along with astounding results from CNN based model, we experimented with a unique DNN-ConvLSTM integrated architecture [91] that combines sequential modeling with convolutional architecture. An LSTM-like algorithm called ConvLSTM uses convolution operations for both input-to-state and state-to-state transitions. In spatio-temporal prediction challenges, it has demonstrated state-of-the-art performance. The DNN-ConvLSTM model's architecture follows a framework similar to our conventional DNN-Single design but with ConvLSTM layers added. Following the first hidden layer, the 512-dimensional features are reshaped into a three-dimensional feature matrix with dimensions (64, 8, 1), which is then used as the input states for a $1D$ ConvLSTM layer. This BN layer is coupled to the ConvLSTM layer, which has 40 kernel filters with 3-length each. This ConvLSTM layer is coupled to a BN layer and has 40 kernel filters, each having a length of 3, which can be noticed from Figure 7.

The model is an ensemble architecture combining a DNN and ConvLSTM layers to tackle fully connected and spatiotemporal feature extraction. The input layer accepts a feature vector $s \in \Re^{193}$, giving the input data. The dL 1^{st} transforms into a 512-dimensional space by determining $\pounds_1 = \mu_1 s + \alpha_1$, where $\mu_1 \in \Re^{512\times193}$ and $\alpha_1 \in \Re^{512}$, initialized using the Glorot uniform transform method. The output $\xi_1 = ReLU(\pounds_1)$, where $ReLU(\pounds) = \max(0, \pounds)$, passes through a dropout layer with a rate of 0.4, deactivating 40% of neurons to reduce overfitting. The resulting output is reshaped into a $3D$ tensor of shape

Table 2: The ASVSpoof 2021 DataSet was used to test the performance of the ASVSpoof 2021 DeepFake detection, yielding the precision, recall, F1-score, macro average, weighted average, and accuracy of the system.

Metrics	**k-NN**			**SVM**			**RF**			**GB**		
	Precision	**Recall**	**F1-score**	**Precision**	**Recall**	**F1-score**	**Precision**	**Recall**	**F1-score**	**Precision**	**Recall**	**F1-score**
SPF	0.85	0.97	0.91	0.96	0.97	0.97	0.97	0.95	0.96	0.96	0.95	0.95
BNF	0.97	0.83	0.89	0.97	0.96	0.96	0.95	0.97	0.96	0.95	0.97	0.95
Macro average	0.91	0.90	0.90	0.96	0.96	0.96	0.96	0.96	0.96	0.95	0.95	0.95
Weighted average	0.91	0.90	0.90	0.96	0.96	0.96	0.96	0.96	0.96	0.95	0.95	0.95
Accuracy % (EER %)	89.95 (14.45)			**96.47 (4.02)**			**95.82 (5.95)**			**95.34 (5.22)**		
Metrics	**AdaBoost**			**XGBoost**			**MLP**					
	Precision	**Recall**	**F1-score**	**Precision**	**Recall**	**F1-score**	**Precision**	**Recall**	**F1-score**			
SPF	0.93	0.93	0.93	0.96	0.97	0.96	0.97	0.95	0.96			
BNF	0.93	0.93	0.93	0.97	0.96	0.96	0.95	0.97	0.96			
Macro average	0.93	0.93	0.93	0.96	0.96	0.96	0.96	0.96	0.96			
Weighted average	0.93	0.93	0.93	0.96	0.96	0.96	0.96	0.96	0.96			
Accuracy % (EER %)	92.99 (8.26)			**96.47 (3.27)**			**95.97 (6.04)**					

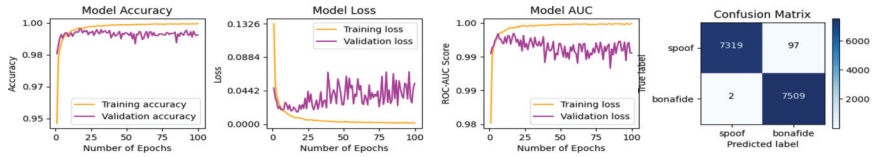

Figure 10: With a confusion matrix applied to the DL-based DNN-BiLSTM method to the dataset, the graphical representation shows the accuracy, loss, and ROC-AUC score.

$(64, 8, 1)$, preparing it for spatiotemporal processing in the ConvLSTM branch. The 1^{st} ConvLSTM1D layer, which applies 40 filters with a kernel size of 3 and '$same'$ padding, preserving temporal dependencies. The ConvLSTM processes the sequence step by step, producing a $3D$ output tensor with the same shape. BN follows stabilizing activations and accelerating convergence. The 2^{nd} ConvLSTM1D layer with 40 filters and '$same'$ padding refines the temporal-spatial feature maps, with BN applied again to ensure stable training. The final output of the ConvLSTM layers is flattened into a $1 - D$ vector, $\xi_{flattened} \in \Re^n$, where n is the total number of features extracted by the ConvLSTM layers.

The flattened vector is processed by a dL with 256 neurons, evaluated as $\pounds_2 = \mu_2 \xi_{flattened} + \alpha_2$, where $\mu_2 \in \Re^{256 \times n}$ and $\alpha_2 \in \Re^{256}$, ReLU activation is applied, and a dropout layer with a rate of 0.5 is added for regularization. In parallel, the DNN processes the input through additional dLs, transforming the feature space sequentially to $256, 128$, and 64 dimensions. Each dL applies $\pounds_i = \mu_i \xi_{i-1}$, and $\xi_i = ReLU(\pounds_{i-1})$, where μ_i and α_i are trainable parameters. Finally, the output of the dLs and ConvLSTM branch are combined, and the final dL determining $\pounds_{final} = \mu_{final} \xi_{combined} + \alpha_{final}$, where $\mu_{final} \in \Re^{2 \times 64}$ and $\alpha_{final} \in \Re^2$. A softmax activation $softmax(\pounds_i) = \frac{exp(\pounds_i)}{\sum_{j=1}^{J} exp(\pounds_j)}$, generates probabilities for the two target classes.

6 Experimental Results and Analysis

In k-NN model, $k = 3$, the algorithm will categorize new instances according to the class labels of the 3 nearest neighbors. From Table 2, the accuracy of k-NN is astounding at 90%, and its precision is 0.85 for SPF and 0.97 for BNF. Recall and F1 score are further parameters for assessing the performance of the k-NN model. The model successfully identified 90% of the actual positive instances, as evidenced by the 97% recall. The 91% F1-score, which balances accuracy and recall, demonstrated great overall performance. The macro average and weighted average metrics demonstrated consistent performance across all classes. The confusion matrix from Figure 8 shows less misclassification for both voice types and strong overall performance.

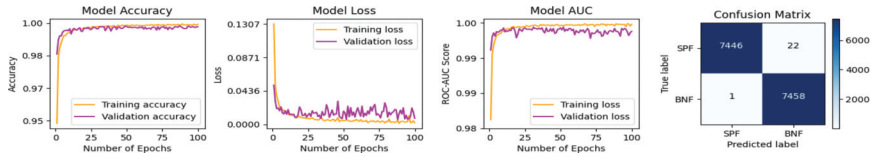

Figure 11: The ROC-AUC score, accuracy, loss, and different confusion matrix from the DL-based DNN-CNN model applied to the dataset are shown graphically.

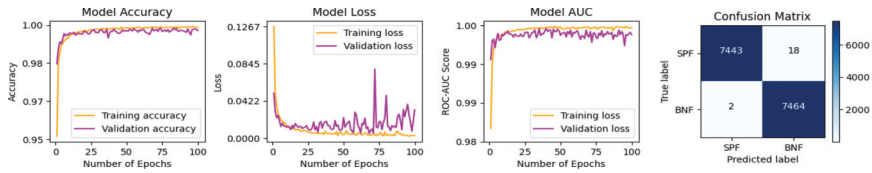

Figure 12: Using another confusion matrix used to apply the DL-based DNN-ConvLSTM method to the dataset, the graphical representation shows the accuracy, loss, and ROC-AUC score.

The SVM algorithm is evaluated and can be noticed from Table 2; the results show that it has an accuracy of 96% with a precision of 0.96 and 0.97 for SPF and BNF, respectively. The recall and F1 scores show outstanding results, with more than 96% in both cases. The macro and weighted average metrics determine more than 96% across all the classes. Additionally, the confusion matrix notices the description from Figure 8, illustrating the rarity of misclassification across both voice types, with notably standing out with the best outcome.

The dataset has employed the RF algorithm, as seen by the Table 2 results. RF has a 95% accuracy rate, which is remarkable. For SPF and BNF, the accuracy is 0.97 and 0.95, respectively. The F1 score is 96% for both cases, and the model correctly displays the recall with 0.95 for SPF and 0.97 for BNF. For RF, the overall macro and weighted average metrics are 0.96. Additionally, the confusion matrix from Figure 8 indicates that both voice types and overall performance have fewer misclassification.

The GB algorithm has been used in the dataset, and the results show that from Table 2. The accuracy is outstanding at 95%. The precision is 0.96 and 0.95 for SPF and BNF, respectively. The model successfully illustrates the recall with 0.95 for SPF and 0.97 for BNF and the F1 score is 95% for both cases. The macro and weighted average metrics for GB are 0.95 across all the cases. Additionally, the confusion matrix in Figure 8, shows that there is less misclassification for both voice types and overall performance is good. AdaBoost Algorithm, one of the ML algorithms, is applied to the dataset, and the results are shown in Table 2, where it is shown that Adaboost has a good accuracy rate at 93%. For each scenario, the precision is 0.93. The

Table 3: The summarising of the replay attack countermeasure methods and their respective results. The results are compared with the hybrid combination of features along with ML and DL models used for the evaluation in terms of EER.

Dataset	Publications	Features
ASVSpoof 2015	Janicki [33]	MFCCs, LFCCs
	Wu *et al.* [87]	Filterbank features, RPS
	Novoselov *et al.* [61]	MFCCs, MFPC, CosPhasePC, MWPC
ASVSpoof 2017	Ji *et al.* [34]	CQCCs
	Adiban *et al.* [2]	MFCCs, RASTA-PLP, CQCCs, LPCCs
	Shim *et al.* [72]	Noise Detection
	Wickramasinghe *et al.* [84]	FDLP (TC and RC)
	Alam *et al.* [3]	DFT-based features_feature normalization
	Sailor *et al.* [70]	ConvRBM-CC
	Lavrentyeva *et al.* [45]	Max-Feature-Map Activation
ASVSpoof 2019	Das *et al.* [17]	Long range acoustic features
	Alzantot *et al.* [5]	MFCCs, CQCCs, STFT
	Chettri *et al.* [15]	Spectral features
	Li *et al.* [51]	MFCCs, CQCCs, Fbank
	Cai *et al.* [10]	IMFCCs, STFT, GD Gram, Joint Gram
	Lavrentyeva *et al.* [46]	CQT, LFCC and DCT
ASVSpoof 2021	Lavrentyeva *et al.* [46]	LFCCs
	Tak *et al.* [77]	LFB
	Lei *et al.* [49]	LFCCs
	Li *et al.* [52]	CQT
	Wang and Yamagishi [81]	LFCCs
	Hua *et al.* [30]	Raw-audio waveform
ASVSpoof 2021	**Our proposed Methods**	**MFCCs, Mel-spectrograms, Chromagram Spectral contrast and Tonnetz features**

Table 3: Continued.

Dataset	Methods	EER
ASVSpoof 2015	AdaBoost	0.158
	DNN-FIT	0.058
	SVM	1.965
ASVSpoof 2017	GMM mean supervector-GB	10.8
	i-vector+GMM, MLP and SVM	10.31
	Neural Network and Multi-task learning	9.56-13.57
	GMM and CNN	9.70
	PCA	11.9
	GMM	8.89
	CNN	6.37
ASVSpoof 2019	DNN	5.95
	ResNet	2.78
	i-vector+Deep and Shallow Classifiers	5.43
	BU	0.67
	ResNet	0.66
	LCNN	0.54
ASVSpoof 2021	LCNN	5.06
	ResNet18-GAT-S	4.48
	Siamese CNN	3.79
	MLCG-Res2Net50+CE	2.15
	LCNN-LSTM-Sum	1.92
	Res-TSSDNet	1.64
ASVSpoof 2021	**KNN, AdaBoost, RF, GB, SVM**	**14.45-4.02**
	MLP, XGBoost, DNN-BiLSTM, DNN-ConvLSTM	**6.04, 3.27, 1.69, 0.71**
	DNN-FIT, DNN-CNN	**0.54, 0.4**

Table 4: Performance of the ASVspoof 2021 Data Set with the Deep Learning Algorithms DNN Single, DNN - CNN, DNN - BiLSTM, and DNN - ConvLSTM.

Metrics	DNN Single	DNN-CNN	DNN-BiLSTM	DNN-ConvLSTM
Precision	0.996	0.997	0.992	0.995
Recall	1.00	0.999	0.999	1.00
Test Loss	0.00585	0.01162	0.02351	0.02131
Accuracy % **(EER %)**	99.798 (**0.54**)	99.825 (**0.40**)	99.569 (**0.71**)	99.778 (**1.69**)
AUC	99.9288	99.8787	99.749	99.8785

recall and F1 scores are 0.93 for both cases. In all, AdaBoosting has a macro and weighted average metric of 0.93. The confusion matrix from Figure 8, shows a reduced rate of misclassification for both speech types and excellent performance.

The XGboost algorithm is applied to the dataset, and the results show that from Table 2. The accuracy of XGBoost is an astounding 96%. The precision is 0.96 and 0.97 for SPF and BNF, respectively. The recall is 0.97 for SPF and 0.96 for BNF. The F1 score is 0.96 for both cases. The macro and weighted average for both cases is 0.96. The confusion matrix from Figure 8 shows that there is less misclassification for both voice types, and overall, the outcomes exhibit strong performance. The MLP algorithm is applied to the dataset, and the results are shown in Table 2. It has an accuracy of 96% and the precision is 0.97 and 0.95 for SPF and BNF, respectively. The recall values for SPF are 0.95 and 0.97 for BNF, while the F1 score is achieved with 96% for both cases. The macro and weighted average for both cases is 0.96. The confusion matrix from Figure 8, shows there is less misclassification in both the voice types, and the overall outcome shows strong performance.

When the dataset is applied for DL, a DNN Single algorithm is used, and the results are shown in Table 4. The precision is 99% while the accuracy of DNN Single is 99%. The AUC score is 0.999. The test loss is just about 0.005. In the confusion matrix for DNN-Single, as shown in Figure 9 among all the matrices, the results stand out particularly well. The DNN-CNN algorithm is subsequently applied to the dataset, and the results are shown in Table 4 with accuracy and precision of more than 99%. The test loss is only 0.011, and the AUC score value is 0.998. Strong performance is shown by the Confusion matrix from Figure 11, for DNN-CNN among all the measures.

The DNN-BiLSTM algorithm is applied to the dataset and it displays the result in Table 4, where the accuracy is 99.8% and precision is 99.1%. The AUC score value is at 0.997, and the test loss is just 0.025. The confusion matrix in Figure 10, for DNN-BiLSTM among all the metrics, the results show good performance. The data set is subjected to the DNN-convLSTM algorithm, and the results show a remarkable precision of 99.7% and precision of 99.5%. The loss is just 0.021. The AUC score value is 0.998. The confusion

matrix from Figure 12, for DNN-convLSTM, shows an astounding performance. The above study is compared with the existing state-of-the-art methods as shown in Table 3. These approaches are investigated in a detailed analysis that will help provide a comprehensive comparison and a deeper understanding of how the methods perform relative to each other. The proposed methods using the hybrid combination of features improved the overall performance of the system. The executable DeepFake code.[3]

DNN Single, DNN-CNN, DNN-BiLSTM, and DNN-ConvLSTM are well-optimized for speech utterances and perform exceptionally well. Introducing Transformer and reinforcement learning models, while powerful in image and multimodal domains, would likely result in unnecessary complexity, increased computational cost, and minimal performance gain for this specific task. The above models leverage the key strengths of speech processing (spectrogram features, sequential dependencies), making these newer models redundant in this context.

7 Conclusion and Future Directions

In this study, we examined both ML and DL. In ML, SVM, RF, GB, Adaboost, XGboost, MLP, and k-NNs are used to identify the SPF and BNF by taking into account the provided spoken utterances. By merging all ML algorithms, we found that SVM and XGBoost showed astounding performance with 96% accuracy by applying the ASVspoof 2021 dataset. In the case of DL, DNN-single, DNN-CNN, DNN-BiLSTM, and DNN-convLSTM, the accuracy rate in all the algorithms is more than 99%. Hence, by combining both DL shows more precision and accuracy rate for detecting spoofs in speech recognition. This resulted in better performance compared to other algorithms.

In our future work, more attention should be paid to the new database source of *ASVspoof Multilingual Librispeech* (MLS). The MLS dataset is divided into distinct subsets to aid in the development of CM, ASV, and SASV systems as well as text-to-speech (TTS) and VC models. We explicitly demonstrated that conventional CNN-based and RNN-based models outperform transformers in low-data settings an observation consistent with outcomes from the ASVSpoof challenges there remains scope for further exploration. Notably, recent top-performing systems, such as those in the ASVSpoof challenge, have favoured CNN-based methods due to their computational efficiency and stability in spoof detection. However, given the rapid evaluation of transformer architectures and their success in other domains, future work could involve a comprehensive analysis incorporating advanced transformer variants. This would help assess their viability for spoof detection tasks under various data

[3] https://github.com/Adityahulk/DeepFake.

availability scenarios and contribute to a more holistic understanding of model performance trade-offs.

The Python-based executable code can be accessed through the DeepFake link.

References

[1] H. Abdullah, K. Warren, V. Bindschaedler, N. Papernot, and P. Traynor, "Sok: The faults in our asrs: An overview of attacks against automatic speech recognition and speaker identification systems", in *2021 IEEE symposium on security and privacy (SP)*, IEEE, 2021, 730–47.

[2] M. Adiban, H. Sameti, N. Maghsoodi, and S. Shahsavari, "Sut system description for anti-spoofing 2017 challenge", in *Proceedings of the 29th Conference on Computational Linguistics and Speech Processing (ROCLING 2017)*, 2017, 264–75.

[3] M. J. Alam, G. Bhattacharya, and P. Kenny, "Boosting the Performance of Spoofing Detection Systems on Replay Attacks Using q-Logarithm Domain Feature Normalization.", in *Odyssey*, Vol. 2018, 2018, 393–8.

[4] M. Aljasem, A. Irtaza, H. Malik, N. Saba, A. Javed, K. M. Malik, and M. Meharmohammadi, "Secure automatic speaker verification (sasv) system through sm-altp features and asymmetric bagging", *IEEE Transactions on Information Forensics and Security*, 16, 2021, 3524–37.

[5] M. Alzantot, Z. Wang, and M. B. Srivastava, "Deep residual neural networks for audio spoofing detection", *arXiv preprint arXiv:1907.00501*, 2019.

[6] L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaría, M. A. Fadhel, M. Al-Amidie, and L. Farhan, "Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions", *Journal of big Data*, 8, 2021, 1–74.

[7] G. K. Birajdar and M. D. Patil, "Speech/music classification using visual and spectral chromagram features", *Journal of Ambient Intelligence and Humanized Computing*, 11, 2020, 329–47.

[8] D. Božić-Štulić, Ž. Marušić, and S. Gotovac, "Deep learning approach in aerial imagery for supporting land search and rescue missions", *International Journal of Computer Vision*, 127(9), 2019, 1256–78.

[9] M. Bubashait and N. Hewahi, "Urban Sound Classification Using DNN, CNN & LSTM a Comparative Approach", in *2021 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT)*, IEEE, 2021, 46–50.

[10] W. Cai, H. Wu, D. Cai, and M. Li, "The DKU replay detection system for the ASVspoof 2019 challenge: On data augmentation, feature representation, classification, and fusion", *arXiv preprint arXiv:1907.02663*, 2019.

[11] J. P. Campbell, "Speaker recognition: A tutorial", *Proceedings of the IEEE*, 85(9), 1997, 1437–62.

[12] G. O. Campos, A. Zimek, J. Sander, R. J. Campello, B. Micenková, E. Schubert, I. Assent, and M. E. Houle, "On the evaluation of unsupervised outlier detection: measures, datasets, and an empirical study", *Data mining and knowledge discovery*, 30, 2016, 891–927.

[13] X. Chen and J. Cheng, "Deep neural network acoustic modeling for native and non-native Mandarin speech recognition", in *The 9th International Symposium on Chinese Spoken Language Processing*, IEEE, 2014, 6–9.

[14] B. Chettri, E. Benetos, and B. L. Sturm, "Dataset Artefacts in anti-spoofing systems: a case study on the ASVspoof 2017 benchmark", *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28, 2020, 3018–28.

[15] B. Chettri, D. Stoller, V. Morfi, M. A. M. Ramírez, E. Benetos, and B. L. Sturm, "Ensemble models for spoofing detection in automatic speaker verification", *arXiv preprint arXiv:1904.04589*, 2019.

[16] A. Chintha, B. Thai, S. J. Sohrawardi, K. Bhatt, A. Hickerson, M. Wright, and R. Ptucha, "Recurrent convolutional structures for audio spoof and video deepfake detection", *IEEE Journal of Selected Topics in Signal Processing*, 14(5), 2020, 1024–37.

[17] R. K. Das, J. Yang, and H. Li, "Long Range Acoustic Features for Spoofed Speech Detection.", in *Interspeech*, 2019, 1058–62.

[18] L. Deng, J. Li, J.-T. Huang, K. Yao, D. Yu, F. Seide, M. Seltzer, G. Zweig, X. He, J. Williams, *et al.*, "Recent advances in deep learning for speech research at Microsoft", in *2013 IEEE international conference on acoustics, speech and signal processing*, IEEE, 2013, 8604–8.

[19] S. Dey, S. Barman, R. K. Bhukya, R. K. Das, B. Haris, S. M. Prasanna, and R. Sinha, "Speech biometric based attendance system", in *2014 twentieth national conference on communications (NCC)*, IEEE, 2014, 1–6.

[20] S. K. Ergünay, E. Khoury, A. Lazaridis, and S. Marcel, "On the vulnerability of speaker verification to realistic voice spoofing", in *2015 IEEE 7th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, IEEE, 2015, 1–6.

[21] F. Eyben, M. Wöllmer, and B. Schuller, "Opensmile: the munich versatile and fast open-source audio feature extractor", in *Proceedings of the 18th ACM international conference on Multimedia*, 2010, 1459–62.

[22] R. Font, J. M. Espín, and M. J. Cano, "Experimental analysis of features for replay attack detection-results on the ASVspoof 2017 Challenge.", in *Interspeech*, 2017, 7–11.

[23] Y. Gao, J. Lian, B. Raj, and R. Singh, "Detection and evaluation of human and machine generated speech in spoofing attacks on automatic speaker verification systems", in *2021 IEEE Spoken Language Technology Workshop (SLT)*, IEEE, 2021, 544–51.

[24] A. Gomez-Alanis, J. A. Gonzalez-Lopez, S. P. Dubagunta, A. M. Peinado, and M. M. Doss, "On joint optimization of automatic speaker verification and anti-spoofing in the embedding space", *IEEE Transactions on Information Forensics and Security*, 16, 2020, 1579–93.

[25] A. Graves, A.-r. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks", in *2013 IEEE international conference on acoustics, speech and signal processing*, Ieee, 2013, 6645–9.

[26] A. Hamza, A. R. R. Javed, F. Iqbal, N. Kryvinska, A. S. Almadhor, Z. Jalil, and R. Borghol, "Deepfake Audio Detection via MFCC Features Using Machine Learning", *IEEE Access*, 10, 2022, 134018–28.

[27] J. H. Hansen and T. Hasan, "Speaker recognition by machines and humans: A tutorial review", *IEEE Signal processing magazine*, 32(6), 2015, 74–99.

[28] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups", *IEEE Signal processing magazine*, 29(6), 2012, 82–97.

[29] Y. Ho and S. Wookey, "The real-world-weight cross-entropy loss function: Modeling the costs of mislabeling", *IEEE access*, 8, 2019, 4806–13.

[30] G. Hua, A. B. J. Teoh, and H. Zhang, "Towards end-to-end synthetic speech detection", *IEEE Signal Processing Letters*, 28, 2021, 1265–9.

[31] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift", in *International conference on machine learning*, pmlr, 2015, 448–56.

[32] H. Jahangir, H. Tayarani, S. S. Gougheri, M. A. Golkar, A. Ahmadian, and A. Elkamel, "Deep learning-based forecasting approach in smart grids with microclustering and bidirectional LSTM network", *IEEE Transactions on Industrial Electronics*, 68(9), 2020, 8298–309.

[33] A. Janicki, "Spoofing countermeasure based on analysis of linear prediction error", in *Sixteenth annual conference of the international speech communication association*, 2015.

[34] Z. Ji, Z.-Y. Li, P. Li, M. An, S. Gao, D. Wu, and F. Zhao, "Ensemble Learning for Countermeasure of Audio Replay Spoofing Attack in ASVspoof2017.", in *Interspeech*, 2017, 87–91.

[35] Y.-G. Jiang, Z. Wu, J. Wang, X. Xue, and S.-F. Chang, "Exploiting feature and class relationships in video categorization with regularized deep neural networks", *IEEE transactions on pattern analysis and machine intelligence*, 40(2), 2017, 352–64.

[36] J.-w. Jung, H. Tak, H.-j. Shim, H.-S. Heo, B.-J. Lee, S.-W. Chung, H.-G. Kang, H.-J. Yu, N. Evans, and T. Kinnunen, "Sasv challenge 2022: A spoofing aware speaker verification challenge evaluation plan", *arXiv preprint arXiv:2201.10283*, 2022.

[37] M. R. Kamble, P. A. K. Sai, M. V. S. Krishna, A. T. Patil, R. Acharya, and H. A. Patil, "Speech Demodulation-based Techniques for Replay and Presentation Attack Detection", in *2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, IEEE, 2019, 1545–50.

[38] Y. Keneshloo, T. Shi, N. Ramakrishnan, and C. K. Reddy, "Deep reinforcement learning for sequence-to-sequence models", *IEEE transactions on neural networks and learning systems*, 31(7), 2019, 2469–89.

[39] A. Khan, K. M. Malik, J. Ryan, and M. Saravanan, "Voice Spoofing Countermeasures: Taxonomy, State-of-the-art, experimental analysis of generalizability, open challenges, and the way forward", *arXiv preprint arXiv:2210.00417*, 2022.

[40] S. Khan and T. Yairi, "A review on the application of deep learning in system health management", *Mechanical Systems and Signal Processing*, 107, 2018, 241–65.

[41] T. Kinnunen, M. Sahidullah, M. Falcone, L. Costantini, R. G. Hautamäki, D. Thomsen, A. Sarkar, Z.-H. Tan, H. Delgado, M. Todisco, *et al.*, "Reddots replayed: A new replay spoofing attack corpus for text-dependent speaker verification research", in *2017 IEEE International conference on acoustics, speech and signal processing (ICASSP)*, IEEE, 2017, 5395–9.

[42] P. Korshunov and S. Marcel, "Impact of score fusion on voice biometrics and presentation attack detection in cross-database evaluations", *IEEE Journal of Selected Topics in Signal Processing*, 11(4), 2017, 695–705.

[43] N. Kriegeskorte and T. Golan, "Neural network models and deep learning", *Current Biology*, 29(7), 2019, R231–R236.

[44] P. Kumari and A. K. Jain, "A Comprehensive Study of DDoS Attacks over IoT Network and Their Countermeasures", *Computers & Security*, 2023, 103096.

[45] G. Lavrentyeva, S. Novoselov, E. Malykh, A. Kozlov, O. Kudashev, and V. Shchemelinin, "Audio replay attack detection with deep learning frameworks.", in *Interspeech*, 2017, 82–6.

[46] G. Lavrentyeva, S. Novoselov, A. Tseren, M. Volkova, A. Gorlanov, and A. Kozlov, "STC antispoofing systems for the ASVspoof2019 challenge", *arXiv preprint arXiv:1904.05576*, 2019.

[47] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning", *nature*, 521(7553), 2015, 436–44.

[48] K. A. Lee, A. Larcher, G. Wang, P. Kenny, N. Brümmer, D. Van Leeuwen, H. Aronowitz, M. Kockmann, C. Vaquero, B. Ma, *et al.*, "The RedDots data collection for speaker recognition", in *Interspeech 2015*, 2015.

[49] Z. Lei, Y. Yang, C. Liu, and J. Ye, "Siamese Convolutional Neural Network Using Gaussian Probability Feature for Spoofing Speech Detection.", in *INTERSPEECH*, 2020, 1116–20.

[50] Q. Li, Y. Yang, T. Lan, H. Zhu, Q. Wei, F. Qiao, X. Liu, and H. Yang, "MSP-MFCC: Energy-efficient MFCC feature extraction method with mixed-signal processing architecture for wearable speech recognition applications", *IEEE Access*, 8, 2020, 48720–30.

[51] R. Li, M. Zhao, Z. Li, L. Li, and Q. Hong, "Anti-Spoofing Speaker Verification System with Multi-Feature Integration and Multi-Task Learning.", in *Interspeech*, 2019, 1048–52.

[52] X. Li, X. Wu, H. Lu, X. Liu, and H. Meng, "Channel-wise gated res2net: Towards robust detection of synthetic speech attacks", *arXiv preprint arXiv:2107.08803*, 2021.

[53] X. Liu, X. Wang, M. Sahidullah, J. Patino, H. Delgado, T. Kinnunen, M. Todisco, J. Yamagishi, N. Evans, A. Nautsch, *et al.*, "Asvspoof 2021: Towards spoofed and deepfake speech detection in the wild", *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2023.

[54] J. M. Martín-Doñas and A. Álvarez, "The vicomtech audio deepfake detection system based on wav2vec2 for the 2022 add challenge", in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2022, 9241–5.

[55] A. Mayr, H. Binder, O. Gefeller, and M. Schmid, "The evolution of boosting algorithms", *Methods of information in medicine*, 53(06), 2014, 419–27.

[56] S. Mo, H. Wang, P. Ren, and T.-C. Chi, "Automatic Speech Verification Spoofing Detection", *arXiv preprint arXiv:2012.08095*, 2020.

[57] V. Monga, Y. Li, and Y. C. Eldar, "Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing", *IEEE Signal Processing Magazine*, 38(2), 2021, 18–44.

[58] N. M. Müller, F. Dieckmann, and J. Williams, "Attacker Attribution of Audio Deepfakes", *arXiv preprint arXiv:2203.15563*, 2022.

[59] A. Nautsch, X. Wang, N. Evans, T. H. Kinnunen, V. Vestman, M. Todisco, H. Delgado, M. Sahidullah, J. Yamagishi, and K. A. Lee, "ASVspoof 2019: spoofing countermeasures for the detection of synthesized, converted and replayed speech", *IEEE Transactions on Biometrics, Behavior, and Identity Science*, 3(2), 2021, 252–65.

[60] M. A. Nielsen, *Neural networks and deep learning*, Vol. 25, Determination press San Francisco, CA, USA, 2015.

[61] S. Novoselov, A. Kozlov, G. Lavrentyeva, K. Simonchik, and V. Shchemelinin, "STC anti-spoofing systems for the ASVspoof 2015 challenge", in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2016, 5475–9.

[62] D. Paul, M. Sahidullah, and G. Saha, "Generalization of spoofing countermeasures: A case study with ASVspoof 2015 and BTAS 2016 corpora", in *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, IEEE, 2017, 2047–51.

[63] D. Pedamonti, "Comparison of non-linear activation functions for deep neural networks on MNIST classification task", *arXiv preprint arXiv:1804.02763*, 2018.

[64] M. H. Rahman, M. Graciarena, D. Castan, C. Cobo-Kroenke, M. McLaren, and A. Lawson, "Detecting Synthetic Speech Manipulation in Real Audio Recordings", in *2022 IEEE International Workshop on Information Forensics and Security (WIFS)*, IEEE, 2022, 1–6.

[65] R. Rahmeni, A. B. Aicha, and Y. B. Ayed, "Acoustic features exploration and examination for voice spoofing counter measures with boosting machine learning techniques", *Procedia Computer Science*, 176, 2020, 1073–82.

[66] S. Ramoji, P. Krishnan, and S. Ganapathy, "NPLDA: A deep neural PLDA model for speaker verification", *arXiv preprint arXiv:2002.03562*, 2020.

[67] J.-M. Ren, M.-J. Wu, and J.-S. R. Jang, "Automatic music mood classification based on timbre and modulation features", *IEEE Transactions on Affective Computing*, 6(3), 2015, 236–46.

[68] Y. Ren, H. Peng, L. Li, X. Xue, Y. Lan, and Y. Yang, "Generalized Voice Spoofing Detection via Integral Knowledge Amalgamation", *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2023.

[69] L. Rice, E. Wong, and Z. Kolter, "Overfitting in adversarially robust deep learning", in *International Conference on Machine Learning*, PMLR, 2020, 8093–104.

[70] H. B. Sailor, M. R. Kamble, and H. A. Patil, "Auditory Filterbank Learning for Temporal Modulation Features in Replay Spoof Speech Detection.", in *Interspeech*, 2018, 666–70.

[71] B. Shamasundar and A. Chockalingam, "A DNN architecture for the detection of generalized spatial modulation signals", *IEEE Communications Letters*, 24(12), 2020, 2770–4.

[72] H.-J. Shim, J.-W. Jung, H.-S. Heo, S.-H. Yoon, and H.-J. Yu, "Replay spoofing detection system for automatic speaker verification using multitask learning of noise classes", in *2018 Conference on Technologies and Applications of Artificial Intelligence (TAAI)*, IEEE, 2018, 172–6.

[73] A. Shrestha and A. Mahmood, "Review of deep learning algorithms and architectures", *IEEE access*, 7, 2019, 53040–65.

[74] S. Styawati, A. Nurkholis, A. A. Aldino, S. Samsugi, E. Suryati, and R. P. Cahyono, "Sentiment Analysis on Online Transportation Reviews Using Word2Vec Text Embedding Model Feature Extraction and Support Vector Machine (SVM) Algorithm", in *2021 International Seminar on Machine Learning, Optimization, and Data Science (ISMODE)*, IEEE, 2022, 163–7.

[75] T. Sun, Y. Wang, J. Yang, and X. Hu, "Convolution neural networks with two pathways for image style recognition", *IEEE Transactions on Image Processing*, 26(9), 2017, 4102–13.

[76] P. Swietojanski, A. Ghoshal, and S. Renals, "Convolutional neural networks for distant speech recognition", *IEEE Signal Processing Letters*, 21(9), 2014, 1120–4.

[77] H. Tak, J.-w. Jung, J. Patino, M. Todisco, and N. Evans, "Graph attention networks for anti-spoofing", *arXiv preprint arXiv:2104.03654*, 2021.

[78] H. Tak, M. Kamble, J. Patino, M. Todisco, and N. Evans, "Rawboost: A raw data boosting and augmentation method applied to automatic speaker verification anti-spoofing", in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2022, 6382–6.

[79] S. Tariq, S. Jeon, and S. S. Woo, "Am I a Real or Fake Celebrity? Measuring Commercial Face Recognition Web APIs under Deepfake Impersonation Attack", *arXiv preprint arXiv:2103.00847*, 2021.

[80] M. Wang, S. Lu, D. Zhu, J. Lin, and Z. Wang, "A high-speed and low-complexity architecture for softmax function in deep learning", in *2018 IEEE asia pacific conference on circuits and systems (APCCAS)*, IEEE, 2018, 223–6.

[81] X. Wang and J. Yamagishi, "A comparative study on recent neural spoofing countermeasures for synthetic speech detection", *arXiv preprint arXiv:2103.11326*, 2021.

[82] X. Wang, J. Yamagishi, M. Todisco, H. Delgado, A. Nautsch, N. Evans, M. Sahidullah, V. Vestman, T. Kinnunen, K. A. Lee, *et al.*, "ASVspoof 2019: A large-scale public database of synthesized, converted and replayed speech", *Computer Speech & Language*, 64, 2020, 101114.

[83] X. Wang, Z. Yan, R. Zhang, and P. Zhang, "Attacks and defenses in user authentication systems: A survey", *Journal of Network and Computer Applications*, 188, 2021, 103080.

[84] B. Wickramasinghe, S. Irtza, E. Ambikairajah, and J. Epps, "Frequency Domain Linear Prediction Features for Replay Spoofing Attack Detection.", in *Interspeech*, 2018, 661–5.

[85] H. Wu, S. Liu, H. Meng, and H.-y. Lee, "Defense against adversarial attacks on spoofing countermeasures of ASV", in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2020, 6564–8.

[86] Z. Wu, T. Kinnunen, N. Evans, J. Yamagishi, C. Hanilçi, M. Sahidullah, and A. Sizov, "ASVspoof 2015: the first automatic speaker verification spoofing and countermeasures challenge", in *Sixteenth annual conference of the international speech communication association*, 2015.

[87] Z. Wu, J. Yamagishi, T. Kinnunen, C. Hanilçi, M. Sahidullah, A. Sizov, N. Evans, M. Todisco, and H. Delgado, "ASVspoof: the automatic speaker verification spoofing and countermeasures challenge", *IEEE Journal of Selected Topics in Signal Processing*, 11(4), 2017, 588–604.

[88] J. Yamagishi, X. Wang, M. Todisco, M. Sahidullah, J. Patino, A. Nautsch, X. Liu, K. A. Lee, T. Kinnunen, N. Evans, *et al.*, "ASVspoof 2021: accelerating progress in spoofed and deepfake speech detection", *arXiv preprint arXiv:2109.00537*, 2021.

[89] A. Zeyer, P. Doetsch, P. Voigtlaender, R. Schlüter, and H. Ney, "A comprehensive study of deep bidirectional LSTM RNNs for acoustic modeling in speech recognition", in *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, IEEE, 2017, 2462–6.

[90] Y. Zhang, F. Jiang, and Z. Duan, "One-class learning towards synthetic voice spoofing detection", *IEEE Signal Processing Letters*, 28, 2021, 937–41.

[91] H. Zheng, F. Lin, X. Feng, and Y. Chen, "A hybrid deep learning model with attention-based conv-LSTM networks for short-term traffic flow prediction", *IEEE Transactions on Intelligent Transportation Systems*, 22(11), 2020, 6910–20.

[92] H. Zheng, Z. Yang, W. Liu, J. Liang, and Y. Li, "Improving deep neural networks using softplus units", in *2015 International joint conference on neural networks (IJCNN)*, IEEE, 2015, 1–4.

[93] T. Zhou, Y. Zhao, J. Li, Y. Gong, and J. Wu, "CNN with phonetic attention for text-independent speaker verification", in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, IEEE, 2019, 718–25.