Time Series-based Spoof Speech Detection Using Long Short-term Memory and Bidirectional Long Short-term Memory

Arsalan R. Mirza[1†] and Abdulbasit K. Al-Talabani[2]

[1]Department of Computer Science, Faculty of Science, Soran University, Soran, Kurdistan Region - F.R. Iraq

[2]Department of Software Engineering, Faculty of Engineering, Koya University, Koya KOY45, Kurdistan Region - F.R. Iraq

***Abstract*—Detecting fake speech in voice-based authentication systems is crucial for reliability. Traditional methods often struggle because they cannot handle the complex patterns over time. Our study introduces an advanced approach using deep learning, specifically long short-term memory (LSTM) and bidirectional LSTM (BiLSTM) models, tailored for identifying fake speech based on its temporal characteristics. We use speech signals with cepstral features such as mel-frequency cepstral coefficients (MFCC), constant Q cepstral coefficients, and open-source speech and music interpretation by large-space extraction to directly learn these patterns. Testing on the ASVspoof 2019 Logical Access dataset, we focus on metrics such as min-tDCF, equal error rate, recall, precision, and F1-score. Our results show that LSTM and BiLSTM models significantly enhance the reliability of spoof speech detection systems.**

***Index Terms* – Bidirectional long short-term memory, Constant Q cepstral coefficients, Countermeasure spoofing, Long short-term memory, Mel-frequency cepstral coefficients, Open-source speech and music interpretation by large-space extraction.**

I. Introduction

Automatic Speaker Verification (ASV) (Bai and Zhang, 2021) is a popular biometric method that identifies a person by analyzing their recorded speech. The main idea is that everyone has a unique voice, similar to unique faces, irises, or fingerprints (Kamble et al., 2020).

Spoofing methods, such as voice conversion (VC), replay attacks, impersonation, and artificial speech and can trick devices that use ASV (Wang et al., 2020) (Adiban Sameti and Shehnepoor, 2020). Impersonation means copying someone's voice to access their account. VC changes the sound of a voice without changing what is said, whereas text-to-speech (TTS) creates fake speech. Replay attacks play recordings of a real voice to trick the system (Kinnunen et al., 2020). Combining these spoofing countermeasures with ASV makes the system more resistant to attacks (Wu et al., 2017).

By understanding, the need for effective countermeasures, several anti-spoofing challenges have emerged (Wu et al., 2015) (Kinnunen et al., 2017) (Todisco et al., 2019). To effectively detect spoof attacks, it is crucial to extract accurate data from speech signals. Choosing effective features is key in detecting spoofed speech. Features covering longer time spans across many frames are needed to detect these artifacts effectively (Tian et al., 2017).

A. Proposed Work Contribution and Organization of the Paper

This paper investigates hand-crafted spectro-temporal representations, using deep learning and feature extraction, for detecting spoofed speech. The paper's contributions are (1) investigating well-known features with sequence-based representations, (2) suggesting deep learning with long short-term memory (LSTM) and bidirectional long short-term memory (BiLSTM) for spoofed speech detection instead of traditional methods, and (3) proposing a new feature by converting open-source speech and music interpretation by large-space extraction's (OpenSMILE's) 88 global representations into time series features.

The paper is organized as follows: Section 2 describes previous related work. Section 3 explains the background of ASVspoof 2019 for logical access, the types of attacks in the ASVspoof 2019 dataset, the feature extraction process, and the model used. Section 4 details the deep learning methodology. Sections 5 and 6 discuss the experimental setup, analyze results, and compare the proposed approach to current systems. The conclusion and directions for future research are covered in Section 7.

ARO-The Scientific Journal of Koya University
Vol. XII, No. 2 (2024), Article ID: ARO.11636. 11 pages
Doi: 10.14500/aro.11636
Received: 31 May 2024; Accepted: 23 August 2024
Regular research paper; Published: 12 September 2024
†Corresponding author's e-mail: arsalan.mirza@soran.edu.iq

 http://dx.doi.org/10.14500/aro.11636

II. Literature Review

Spoofing speech detection has become increasingly important because of the growth in deceiving activities such as voice impersonation, conversion, synthetic speech attacks, and deepfake technology. To resolve this, researchers have investigated numerous methods for distinguishing genuine speech from fake speech (Nautsch et al., 2021).

A typical method for spoofing speech detection is extracting the acoustic feature from the speech signal (Rahmeni, Aicha, and Ayed, 2020) (Dave, 2013). The primary focus of early research was on conventional acoustic characteristics, however, the development of machine/deep learning approaches encouraged the researchers to focus on more complex to enhance the deep learning-based features/models (Jiang et al., 2009) (Kumari and Jayanna, 2015) (Ahmed et al., 2022).

Recent studies have used deep learning architectures such as recurrent neural networks (RNNs) and convolutional neural networks (CNN) to extract discriminative features from speech signals. A 1D-CNN+LSTM approach (Ahmed et al., 2022) was proposed on the ASVspoof 2019 dataset, achieving an equal error rate (EER) of 31.9%. A hybrid data augmentation technique using the synthetic minority over-sampling technique was implemented (Chakravarty and Dua, 2023) by employing an LSTM and support vector machine classifier, achieving EERs of 5.1% and 7.4% with 93% and 92% accuracy, respectively. A similar approach was used (Zhou et al., 2022) for the ASVspoof 2019 PA subset, using GTCC and mel-frequency cepstral coefficients (MFCC) features. The BiLSTM network achieved an accuracy of 97% and an EER of 2.97% through consistent implementation across multiple approaches.

Researchers have enhanced the accuracy of identifying fake speech by combining multiple acoustic features. As suggested by Karo, Yeredor, and Lapidot, 2022, they applied new methods based on probability mass function estimation to audio waveforms in the ASVspoof 2019 LA subset. They focused on two types of filter banks (MFCC and GTCC) and used diffusion maps to reduce dimensionality, measuring similarity through diffusion distance. Their evaluation of this subset achieved an EER of 12.09% for males and 12.99% for females. Similarly, Hassan and Javed, 2021 proposed an effective synthetic speech detector by combining spectral features including MFCC, GTCC, spectral flux, and spectral centroid. Their model, trained on 15,981 samples and tested on 14,161 samples, achieved an impressive EER of 3.05%.

Moreover, researchers have explored domain-specific knowledge, such as adopting different speech representations of the front-end model and the fusion of different temporal segments. Another research (Wei, Pang and Kuo, 2024) proposed a Green ASVSpoof detector based on pre-trained speech representations by extracting the probability vectors, probability histograms, and probability patterns of fusion of three XGBoost classification stages and achieved 1.82% EER for the 2019 LA evaluation subset and lead to lower model sizes and inference complexity per input speech sample.

In addition to employing acoustic and deep features, various methods and platforms exist for extracting features from speech signals. For instance, according to Devesh et al., 2022, an 88-dimensional OpenSMILE feature set was applied to LJ speech, CMU-arctic, and LibreTTS datasets.

A. Research Gap

Recent reviews of ASV and countermeasure systems highlight the ongoing need for improvement in this critical area. Both ASV and CM systems traditionally use acoustic and deep features for extracting features. Deep features (non-handcrafted features) are extracted from the multiple layers of a neural network. These features, which do not have explicit, pre-defined meanings, are learned during the training process. Neural networks capture complex patterns and hierarchical structures in the data through these deep features. However, few studies explore the effects of MFCC and constant Q cepstral coefficient (CQCC) features, especially in their time-series formats such as separate, concatenated, and fused forms. In addition, there has been limited focus in related research on time-series representations of speech, possibly due to the complexity of handling large amounts of data. Furthermore, as far as we know, the application of time-series OpenSMILE features with different time intervals has not been explored before.

III. Background

A. ASVspoof 2019 Logical Access Subset

The ASVspoof 2019 project's logical access subset is part of its third version. It includes fake speech made by TTS or VC models. This subset contains 12,483 real utterances and 108,978 fake ones created by 19 different methods, such as 11 TTS techniques, 5 VC techniques, and three hybrid approaches. ASVspoof 2019 covers both logical access and physical access scenarios with a wider variety of spoofing methods and a larger dataset.

The ASVspoof 2019 logical access subset is part of the third version of the ASVspoof project. Spoofed samples are generated using TTS or VC models. The ASVspoof 2019 LA subset comprises 12,483 bonafide utterances and 108,978 spoofed utterances. These spoofed utterances are created using 19 different algorithms, including 11 TTS techniques, five VC techniques, and three hybrid approaches (as detailed in Table I).

The logical access subset of the ASVspoof 2019 dataset is divided into three subsets: training, development, and evaluation. The training subset is used to train spoofing countermeasures, whereas the development subset is used

TABLE I
A Summary of the ASVspoof 2019 Logical Access Dataset

ASVspoof 2019 LA Subset	Speaker		Utterance	
	Male	Female	Bonafide	Spoof
Train	8	12	2,580	22,800
Development	8	12	2,548	22,296
Evaluation	30	30	7,355	63,882

to optimize these countermeasures. Finally, the evaluation subset assesses the performance of the developed models.

The spoofing algorithms that are present in the evaluation data are not present in the training and development subsets. Out of 19 different spoofing algorithms, six (A01–A06) of them have been used for generating the utterance of the train and development subset whereas the remaining 13 (A07–A19) have been used for generating the evaluation subset.

B. MFCC, CQCC, and OpenSMILE Feature

MFCC are crucial features widely used in voice signal processing. The process of extracting MFCC involves dividing the signal into frames, calculating the energy spectrum, applying Mel filter banks, computing logarithms for each filter bank output, and performing discrete cosine transform (Novoselov et al., 2016). The computation and extraction steps are depicted in Fig. 1.

Short-term spectral features such as MFCC are commonly used in speech recognition systems (Abdul and Al-Talabani, 2022). The higher frequency filters in the Mel-scale filter bank used by MFCC have wider bandwidths compared to lower frequency filters, but they maintain the same temporal resolutions (Patel and Patil, 2015).

Studies have shown that CQCC features perform well in utterance and speaker verification (Todisco, Delgado and Evans, 2016). CQCC extraction involves using the constant-Q transform (CQT), which provides enhanced frequency resolution for lower frequencies and improved temporal resolution for higher frequencies (Todisco et al., 2019).

Fig. 2 illustrates how CQCC features are extracted. According to Todisco, Delgado, and Evans (2016), studies have used three different dimensions of CQCC features: 12, 19, and 29, all including C0. In the context of CQCC features, C0 represents the average energy across frequency bands after applying the CQT to the signal. The initial choice of 12 and 19 dimensions is based on their common use in speech and speaker recognition. The 29 dimension aims to explore whether higher coefficients provide additional information useful for detecting spoofing.

In the opposite, Yang, Das and Li, 2020, claimed that the CQT feature, more precisely the log power spectrum of the CQT, does not have the phase information of the signal. To further generate the CQCC, even more information will be discarded. From a hand-crafted feature engineering point of view, a good feature must capture discriminative information between classes and must also be compact in size.

Moreover, there is another feature that has received relatively little attention in the area of research for spoof detection, which is the OpenSMILE feature. Open Speech and Music Interpretation by Large Scale (OpenSMILE) (Eyben, Wöllmer and Schuller, 2010) is an innovative open-source tool designed for extracting features in speech processing and music information retrieval. Its main purpose is to facilitate audio feature extraction. OpenSMILE offers a straightforward, scriptable console application where modular feature extraction components can be easily configured, allowing researchers to take advantage of features across various domains. The OpenSMILE[1] features are an open-source toolkit that extracts essential speech features. OpenSMILE features include three standard support features – ComParE 2016, GeMAPS, and eGeMAPS. ComParE 2016 (Eyben, Wöllmer and Schuller, 2010) is the largest in terms of size and each feature can be extracted in the low-level descriptor (LLD) or functional. The contribution of every feature including MFCC, CQCC, and OpenSMILE has been investigated in detecting fake speech, this motivated us to use the eGeMAPSv02 (Eyben et al., 2016) contains 88 functional parameters. The LLD contains 25 feature-level parameters for each 20 ms and with a hop length of 10 ms.

1. http://www.audeering.github.io/opensmile

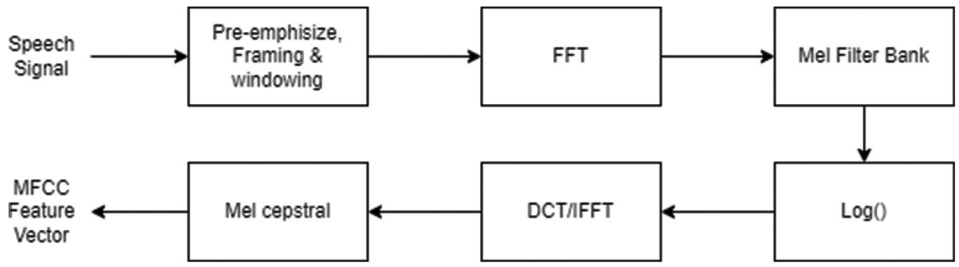

Fig. 1. Mel-frequency cepstral coefficient feature extraction process.

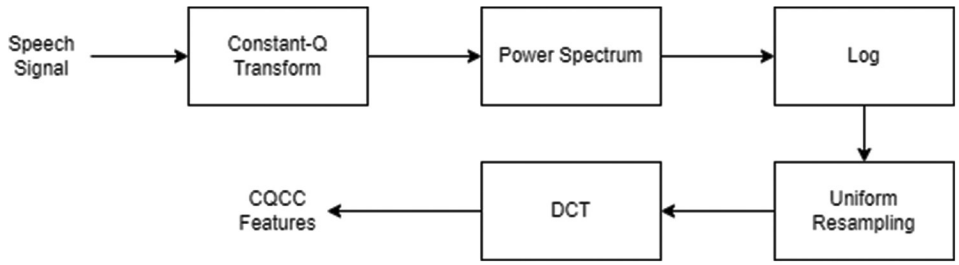

Fig. 2. Constant Q cepstral coefficients feature extraction process.

http://dx.doi.org/10.14500/aro.11636

C. LSTM and BiLSTM

LSTM is a specialized form of RNN widely utilized in natural language processing and time series prediction tasks. Unlike a standard LSTM, where the input flows only in one direction, a BiLSTM processes input in both forward and backward directions, allowing it to capture important contextual information from both past and future states. Fig. 3 illustrates the cell state of an LSTM model, highlighting its internal mechanisms.

The cell state is a crucial component in LSTM networks, responsible for maintaining long-term dependencies and preserving relevant information across extended sequences. Serving as the memory within the LSTM unit, it plays a vital role in capturing and retaining data from previous time steps (illustrated in Fig. 4, left). The cell state is updated and modified through three primary gates: the forget gate, the input gate, and the output gate. These gates work together to determine which information should be remembered or discarded, ensuring the LSTM effectively manages data throughout the sequence.

LSTM is specifically designed to update information across different time steps, overcoming the limitations of traditional RNNs. Unlike RNNs, LSTMs excel at capturing long-term dependencies in sequential data (as shown in Fig. 4, left). This capability makes LSTMs highly effective for time series prediction and other tasks involving sequential data in deep learning architectures. The BiLSTM network, illustrated in Fig. 4, right, enhances this functionality by processing data in both forward and backward directions, thereby integrating information from both past and future time steps for more comprehensive modeling.

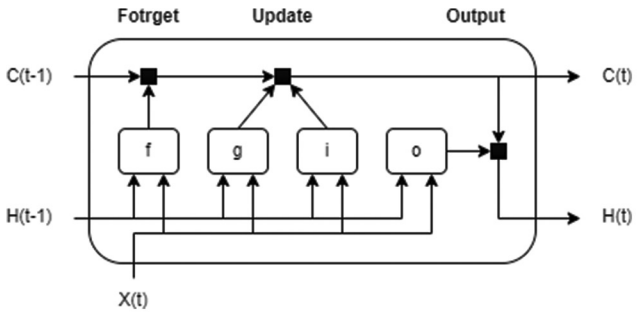

Fig. 3. Cell state used in long short-term memory.

IV. Deep Learning Approach Methodology

This study primarily aimed to create a countermeasure to work alongside the proposed ASV system, which was included in the ASVspoof 2019 challenge. In this section, the architecture of the proposed CM system is presented. Regarding the feature extraction from the speech signal, the paper adopts the use of the CQCC, MFCC, and OpenSMILE features toolkit which will be described in section 4.1. At the classifier level, LSTM (Hochreiter and Schmidhuber, 1997) and BiLSTM are utilized to evaluate the data of the ASVspoof 2019 LA subset to build different CM systems. Fig. 5, shows an illustration of the proposed CM block diagram. The suggested model operated in sequence-based phases and each phase had several processes carrying out distinct tasks.

As illustrated in Fig. 5, the proposed method comprises several tasks of the deep learning model. The model is trained using the ASVspoof 2019 logical access subset. The main components of a spoofing detection system are feature extraction and decision-making modules. In front-end features extraction, time series-based features such as (MFCC, CQCC, and OpenSMILE) have been used, For the back-end speech spoofing countermeasure module, we explore the LSTM and BiLSTM classifiers for training the model, and the posterior probability is used for decision making. In our model the structure is a sequence-to-label classification, we have created a sequence input layer, followed by an LSTM layer, then a fully connected layer, and a Softmax layer at the end.

A. Feature Extraction

In this phase, we will discuss the used features and the extraction process. Feature extraction involves transforming raw speech data into a set of attributes or characteristics that can be used for analysis. This process includes selecting and transforming data to create informative, non-repetitive features that enhance model performance.

MFCC features

Feature extraction aims to provide a clear representation of the vocal tract based on its response characteristics. By leveraging the capabilities of the human auditory system (HAS), MFCC can accurately capture key parameters of speech signals across different voice transformation scenarios.

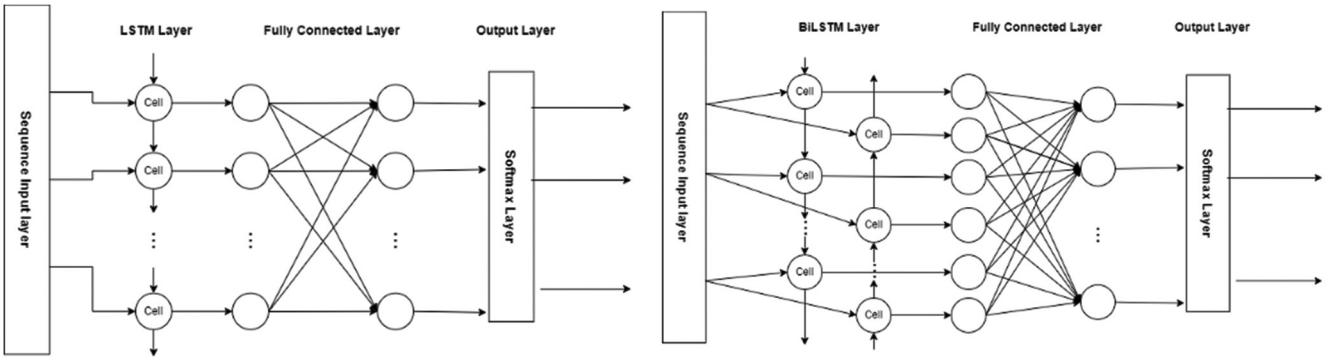

Fig. 4. The structure of long short-term memory (left) and bidirectional long short-term memory (right).

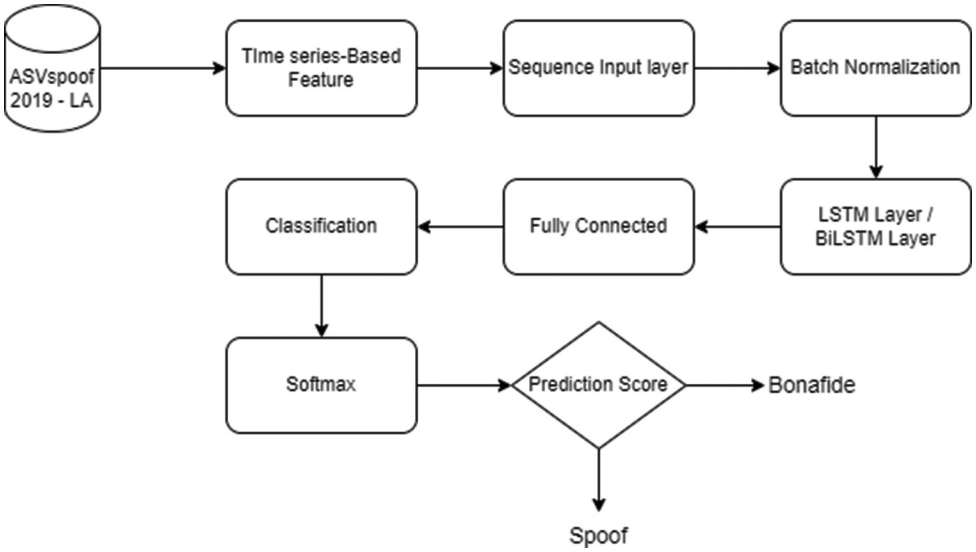

Fig. 5. The structure of the main approach of the proposed countermeasure.

In this study, the Librosa (McFee et al., 2015) library was used to extract a sequence of 12-dimensional MFCC feature vectors with log energy for each spoken sentence. MFCC and log power magnitude spectra (LPMSs) are obtained using a pre-emphasis with a coefficient of 0.97. Each frame is acquired by applying a 30 ms Hamming window with a step size of 15 ms.

CQCC features

CQCC is derived from CQT, which creates a time-frequency representation for speaker recognition and spoofing detection (Todisco, Delgado and Evans, 2016). CQCCs are adept at capturing distinct spectral features across various frequency levels, making them effective in distinguishing between different types of audio signals (Todisco, Delgado and Evans, 2017).

The CQCC feature was extracted with a time step of 12 ms, covering nine octaves with 96 bins per octave. Each feature vector includes 19 cepstral coefficients, including the 0^{th} coefficient, along with their delta and double delta coefficients. Hence, each feature vector is 60-dimensional. In addition, we have extracted another CQCC feature set with different parameters. This new feature is extracted with nine octaves with 171 bins per octave and 12 static cepstral coefficients including 0^{th} coefficient.

OpenSMILE features

The OpenSMILE features are an open-source toolkit that enables to the extraction of essential speech features such as auditory formants spectral, signal energy, MFCC, Shimmer, Jitter, linear predictive coding, pulse code modulation, and line spectral pairs. While the ComParE 2016 and GeMAPS can be used as a global representation of speech samples with 6373 and 88 features, respectively, we have used eGeMAPSv02 with LLDs as a feature set with 25 time-step parameters for each 20 ms and with a hop length of 10 ms.

Using the OpenSMILE feature, the global representation of each speech sample with eGeMAPSv02 which contains 88 functional parameters has been extracted. For a new time, series-based feature we have created a new feature by splitting the speech signal into different frames and extracting the functional features for each frame. For this feature set, the hamming window has a size of 100 ms, and a step size of 40 ms is applied to extract the features.

B. Combined Feature

Combining features in the context of deep learning and data processing is the process of integrating multiple types or sets of features to create a more comprehensive and informative representation of the data. We have concatenated MFCC with CQCC to improve the accuracy and robustness of the system by providing diverse perspectives on the audio signal's characteristics. In this paper, we have used a combined feature set that consists of 13 MFCC and 13 CQCC to construct a new feature set with 26 dimensions of time steps whereas the length of utterance remains the same.

V. Experimental Setup

The implementation of the proposed deep learning approaches is carried out using Python version 3.9.13. We evaluated our models on the ASVspoof 2019 evaluation part for the logical access subset.

The proposed countermeasure models (LSTM and BiLSTM) have been implemented using MATLAB 2023b and Python on the Windows 10 operating system with 64GB of RAM and Processor Intel(R) Core (TM) 3.6GHz 16 CPUs and a Single GPU with 10 GB of size. Data extraction and preprocessing have been conducted using Jupyter Notebook within the Anaconda environment.

A. LSTM and BiLSTM Parameters Tuned

We used 1000 hidden states across all feature types in both models, even though most speech samples have fewer than 1000 time steps per sample. Our approach included training on the ASVspoof 2019 Train subset and testing on the development subset. We tested various layer numbers and configurations to optimize model

performance. Each model was trained for 100 epochs with a fixed learning rate of 0.001 and a batch size of 128 for efficient improvement.

B. Experiments

Table II provides a detailed overview of the feature sets used in our investigation, including the number of features linked to each time step for LSTM and BiLSTM models. This systematic approach enables us to explore the efficiency of these models in extracting the artifact within the different types of features.

During the training of the models, the BiLSTM was slower than the LSTM model. This observation suggests that BiLSTM models can capture additional features in the data. In this regard, Siami-Namini, Tavakoli and Namin, 2019, recommended using BiLSTM instead of LSTM for forecasting problems in time series analysis.

C. Fusion of Models

Fusion can occur at various levels, including feature-level fusion, decision-level fusion, or model-level fusion, depending on the specific application and context. In this set of experiments, model-level fusion, also known as late fusion, was used by combining the results from different models through averaging their outputs. This method allows for the creation of fusion models that use either the same classifier with different features or the same features with different classifiers to improve overall performance. Different model-level fusions were conducted, selecting the best-performing models based on various evaluation criteria to achieve more robust and effective results.

VI. Results and Discussion

In this section, we present the findings of our experimental analysis and engage in a comprehensive discussion of the observed results. We begin by summarizing the outcomes of our model implementations and highlighting key performance metrics.

A. Feature-Level Fusion

Feature-level fusion is used for merging different types of features to enhance the performance of models. By combining multiple feature sets, enables more accurate and robust predictions and leveraging the strengths of each feature set.

TABLE II
Experimental Setup Time Series-Based Feature

Feature ID	Time series-based feature	Dimension
CQCC_60	CQCC	60
CQCC_13	CQCC	13
MFCC_13	MFCC	13
MFCC_CQCC_26	MFCC+CQCC	26
OS_25	OpenSMILE	25
OS_88	OpenSMILE	88

CQCC: Constant q cepstral coefficients, MFCC: Mel-frequency cepstral coefficients

LSTM

With the LSTM model, we conducted the experiments using six different types of time series-based features, without employing any data augmentation process. The results of our analysis, as detailed in Table III, reveal significant insights into the performance of the LSTM models across various feature sets. Notably, the LSTM model with CQCC_60 and OS_25 features outperformed other features in all evaluation metrics.

As shown in Table III, the model with the CQCC with 60 dimensions of time step feature obtained the best EER (6.15%), Min-tDCF (0.1917), and Recall (99.91%) among other types of features. From the OpenSMILE time series-based feature (OS25 and OS88), we can conclude that the higher number of time step features does not guarantee to have a better result.

In experiment 5, with OS (25), an outstanding result was achieved in terms of accuracy, precision, and f1-score whereas the OS (25) has been extracted in the LLD of the OpenSMILE feature[2]. Furthermore, in experiment 6, the OpenSMILE with 88 time series-based features did not perform better than experiment 5 with OpenSMILE 25.

BiLSTM

The result of the same features has been used with the BiLSTM model and the result is shown in Table IV. As we can see, in experiment 9, the 13 MFCCs time series-based features outperform all other features. In addition, this feature achieved the best result in terms of accuracy (93.05%).

The BiLSTM models outperformed their unidirectional LSTM in terms of accuracy, precision, recall, and f1-score whereas on the other hand, the LSTM model obtained outstanding results for EER (6.15%) and min-tDCF (0.1917). While the BiLSTM model is capable of capturing complex temporal dependencies as shown in Table IV, these findings underscore the enhanced capability of BiLSTM models in leveraging bidirectional context (Table IV).

Experiments 4 and 10, where a combination of CQCC and MFCC is performed, did not lead to improving the result of both LSTM and BiLSTM models.

B. Fusion Models Result

In this section, the fusion models focus on combining the best-performing LSTM and/or BiLSTM experiments. A total of 12 experiments were carried out 6 for each model with the top performers identified using various evaluation metrics. Initially, the two best experiments from both LSTM and BiLSTM models were selected, followed by the selection of the best metrics between the two. Eventually, the selection was expanded to include the top three and four experiments for each model. To ensure a comprehensive analysis, experiments were selected from Tables III and IV. An averaging technique was applied to merge the results from the selected experiments in these fusion models.

A fusion model of the top two experiments, 1 and 9, produced the best results (Table V) in terms of EER and

2. https://audeering.github.io/opensmile-python/usage.html#process-signal

TABLE III
Long Short-Term Memory Results with Different Types of Features

Experiment ID	Feature	Equal error rate %	Min-tDCF	Accuracy %	Precision %	Recall %	F1-score %
1	CQCC (60)	6.15	0.1917	80.51	78.33	99.91	87.82
2	CQCC (13)	13.06	0.4476	58.60	54.10	99.51	70.09
3	MFCC (13)	9.35	0.2456	81.60	79.63	99.81	88.59
4	MFCC_CQCC (26)	11.52	0.3475	58.22	53.47	99.88	69.66
5	OS (25)	8.25	0.2292	88.55	87.50	99.69	93.20
6	OS (88)	9.80	0.2344	86.74	85.39	99.78	92.03

CQCC: Constant q cepstral coefficients, MFCC: Mel-frequency cepstral coefficients

TABLE IV
BiLSTM Results with Different Types of Features and Different Time Step

Experiment ID	Feature	Equal error rate %	Min-tDCF	Accuracy %	Precision %	Recall %	F1-score %
7	CQCC (60)	9.74	0.3090	81.87	80.13	99.56	88.80
8	CQCC (13)	21.46	0.6325	62.03	58.60	98.42	73.46
9	MFCC (13)	6.29	0.1937	93.05	92.71	99.50	95.99
10	MFCC_CQCC (26)	11.02	0.3748	82.25	80.90	99.14	89.10
11	OS (25)	8.93	0.2380	89.39	88.62	99.49	93.74
12	OS (88)	8.33	0.2333	87.99	86.89	99.68	92.84

CQCC: Constant q cepstral coefficients, MFCC: Mel-frequency cepstral coefficients

TABLE V
Fusion Model Result of Combining Different Types of Experiments

Fusion models	Equal error rate %	Min-tDCF	Accuracy %	Precision %	Recall %	F1-score %
1,4	4.28	0.17766	85.66	84.05	99.94	91.31
1,5	5.54	0.19461	88.32	87.03	99.93	93.03
1,9	4.06	0.15862	91.64	90.75	99.92	95.11
1,12	5.88	0.19494	88.45	87.20	99.91	93.12
5,6	8.26	0.21688	88.00	86.78	99.81	92.84
5,9	5.98	0.18099	92.04	91.32	99.79	95.37
7,12	8.71	0.21638	87.33	85.99	99.86	92.41
9.11	6.25	0.19408	92.21	91.60	99.69	95.47
9.12	6.07	0.18543	91.66	90.90	99.78	95.13
1,3,4	4.73	0.18489	77.03	74.41	99.97	85.31
1,3,5	4.86	0.17893	87.01	85.56	99.95	92.19
1,5,6	6.06	0.19496	89.08	87.97	99.84	95.53
3,5,6	7.39	0.20877	87.46	86.13	99.87	92.49
7,9,12	6.02	0.19064	88.65	87.45	99.88	93.25
9,11,12	6.11	0.20110	89.73	88.77	99.75	93.94
1,3,5,6	5.18	0.19382	87.53	86.14	99.95	92.53
7,9,11,12	6.02	0.19872	88.99	87.83	99.87	93.46
1,3,5,6,7,8,11,12	5.87	0.19818	87.71	86.34	99.93	92.64

min-tDCF. The obtained results are 0.15862 for min-tDCF and 4.06% of EER. Consequently, in experiments 9 and 11, the best fusion model improved its accuracy and precision to 92.21% and 91.60%, respectively.

Based on the findings in Table V, the fusion model that combines CQCC and MFCC features (specifically fusion [1,4] and fusion [1,9]) shows that using the fusion of these features results in improved performance, particularly by achieving lower EER and min-tDCF values. It can also be inferred that combining different types of features and classifiers enhances the model's overall performance. However, unlike the EER, the fusion models did not outperform the single model in terms of accuracy (93.05%) and precision (95.99%) when compared to the single model in experiment 9.

C. Attack-Based Analysis Result

The ASVspoof 2019 logical access dataset comprises 13 unseen attacks within the evaluation subset. The attack-based analysis section aims to identify the attack type with the most noticeable effect on the overall result. Within the LA subset, attack types range from A07 to A19. Each attack type consists of a total of 4914 samples, whereas 7355 samples are categorized as bonafide.

Fig. 6 offers insights into the performance of both the LSTM (left) and BiLSTM (right) models utilizing CQCC (60) in detecting spoofs created by various attack types (including A07, A09, A16, and A19) within the LA subset, showcasing the percentage of missed samples for each attack type.

The LSTM model demonstrates superior performance in detecting bonafide, A07, A09, A12, A16, and A19. In contrast, the BiLSTM model surpasses the LSTM in identifying attacks A10, A11, A14, and A15, yielding better results. However, it also struggles with misclassifications for attacks A13, A17, and A18.

Fig. 7 shows that utilizing CQCC with 13 dimensions did not yield improved results when compared to using CQCC with 60 dimensions. Despite the lower dimensionality, the performance of both LSTM and BiLSTM models did not significantly improve. Interestingly, among the attacks, only A07, A16, and A19 were consistently identified correctly by both models, irrespective of the dimensionality of the CQCC feature. This suggests that while reducing the dimensionality of the feature may offer computational advantages, it does not necessarily enhance the models' ability to accurately classify certain attack types.

Fig. 7. Miss classification of samples for each attack type of constant Q cepstral coefficients feature of 13 dimensions with long short-term memory (left) and bidirectional long short-term memory (right) models.

Even though the BiLSTM model demonstrated a higher misclassification rate of bonafide samples at 3.97%, the LSTM model showcased improved performance with a lower misclassification rate of 1.27%, as illustrated in Fig. 8. Moreover, it is noteworthy that despite this difference, the BiLSTM model exhibited superior performance in detecting all attack types when utilizing MFCC features, outperforming the LSTM models employing the same features. This suggests that while the LSTM model may excel in certain aspects, such as accurately classifying bonafide samples, the BiLSTM model shows promise in overall attack detection when leveraging MFCC features.

Fig. 8. Miss classification of samples for each attack type of mel-frequency cepstral coefficients feature of 13 dimensions with long short-term memory (left) and bidirectional long short-term memory (right) model.

In the LSTM model (depicted in Fig. 9), although the results differ from those of the BiLSTM model for attack types A07 to A16, the misclassification rate of bonafide samples is lower with LSTM at 0.53% compared to the BiLSTM models.

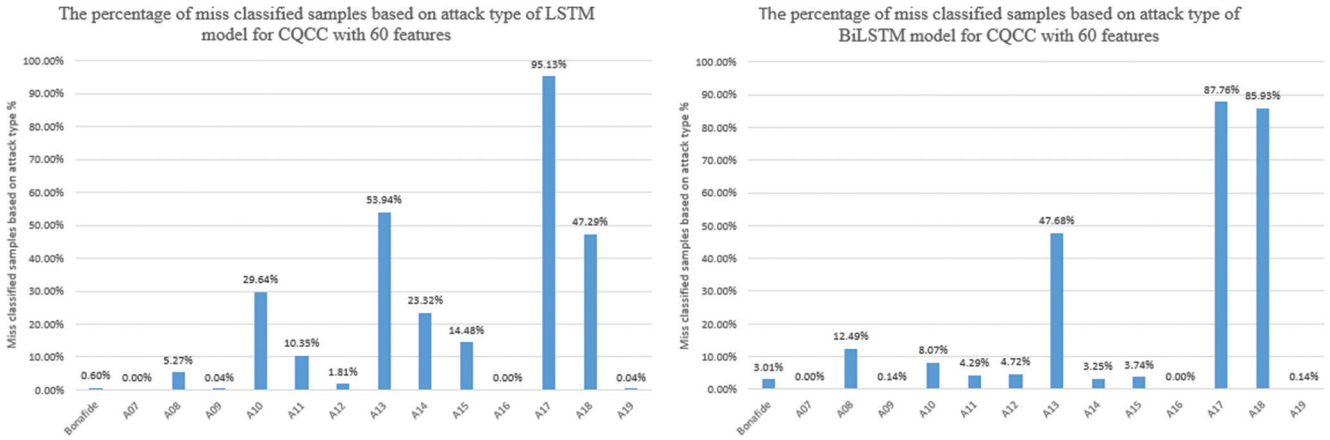

Fig. 6. Miss classification of samples for each attack type of constant Q cepstral coefficients feature of 60 dimensions with long short-term memory (left) and bidirectional long short-term memory (right) models.

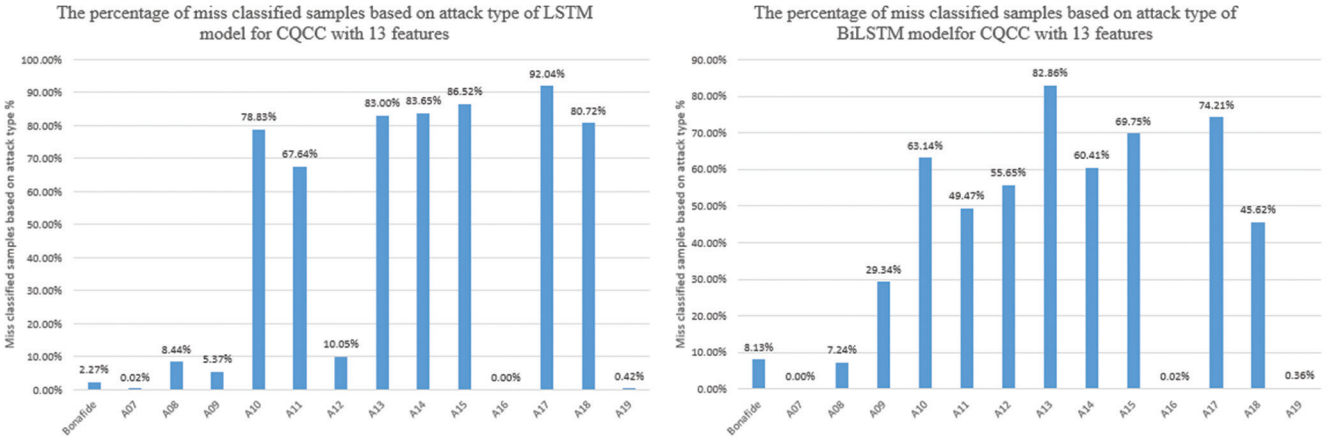

Fig. 7. Miss classification of samples for each attack type of constant Q cepstral coefficients feature of 13 dimensions with long short-term memory (left) and bidirectional long short-term memory (right) models.

Fig. 10 shows the misclassification outcomes based on attack types using OpenSMILE with 25 time series-based features. While both the LSTM and BiLSTM models demonstrate similar overall performance, they both notably enhance the identification of A07, A09, A10, A11, A12, A13, A14, A15, and A16 attack types. However, both models exhibit a significantly higher rate of misclassification for A18, A17, A08, and A19 attacks, as well as bonafide samples.

Fig. 11 illustrates the percentage of misclassified samples by both LSTM and BiLSTM models across various attack

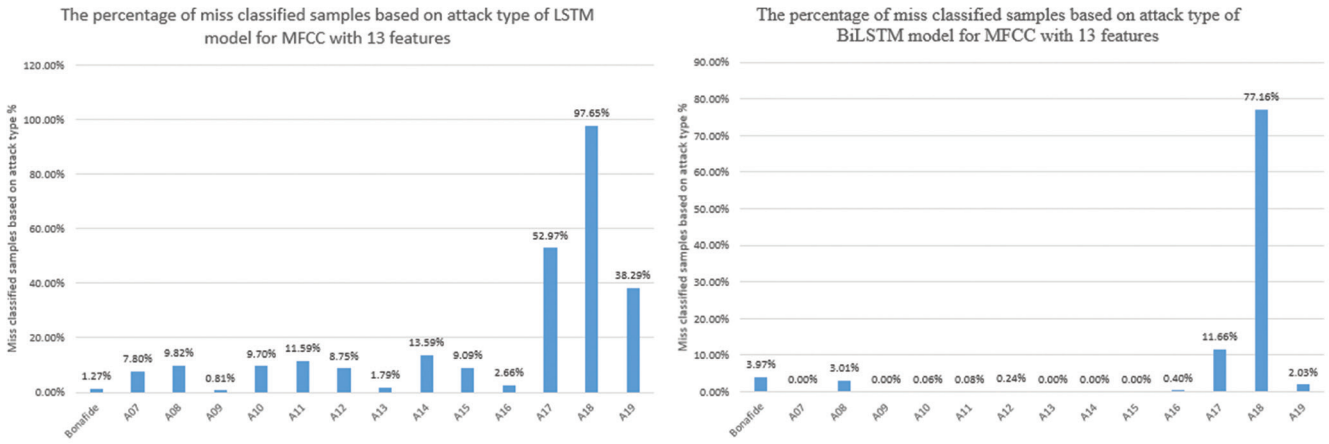

Fig. 8. Miss classification of samples for each attack type of mel-frequency cepstral coefficients feature of 13 dimensions with long short-term memory (left) and bidirectional long short-term memory (right) model.

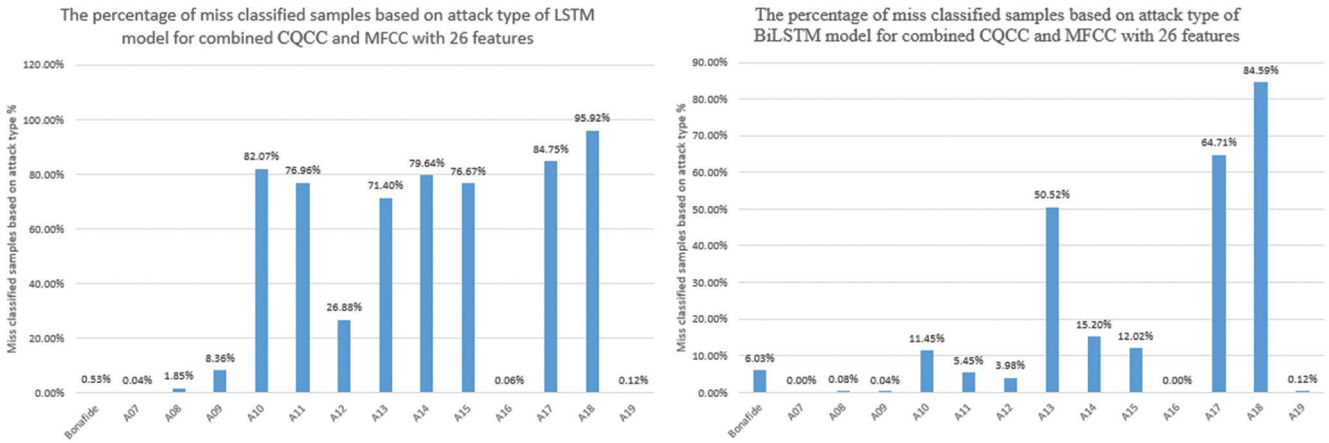

Fig. 9. Miss classification of samples for mel-frequency cepstral coefficients and constant Q cepstral coefficients combined features of 26 dimensions with long short-term memory (left) and bidirectional long short-term memory (right) model.

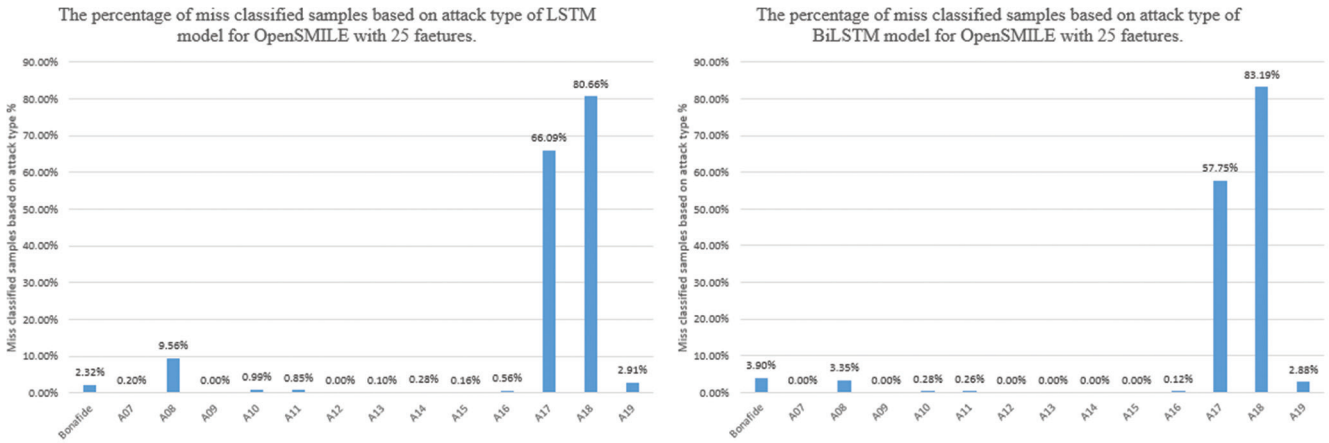

Fig. 10. Miss classification of samples for open-source speech and music interpretation by large-space extraction of 25 dimensions with long short-term memory (left) and bidirectional long short-term memory (right) model.

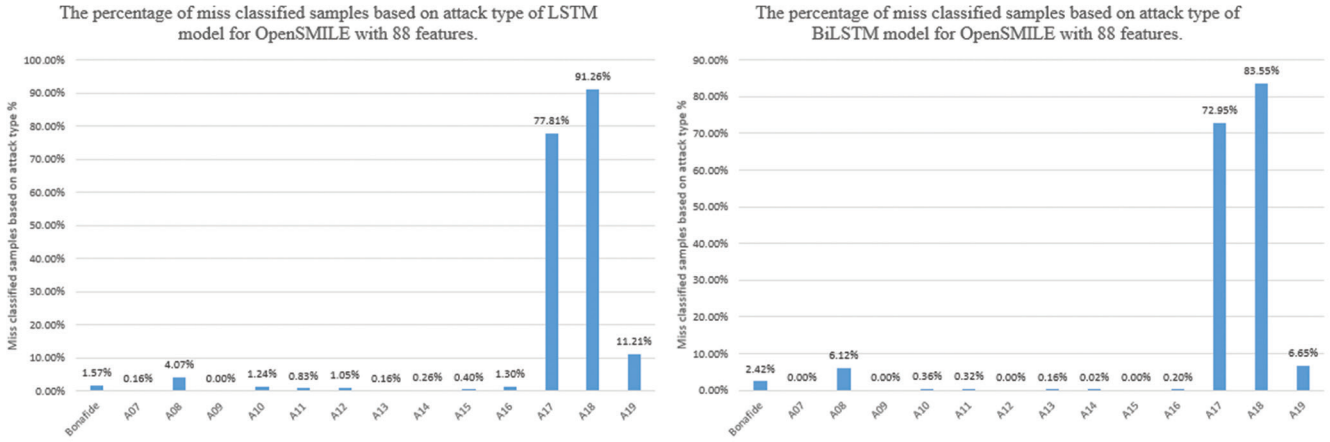

Fig. 11. Miss classification of samples for open-source speech and music interpretation by large-space extraction of 88 dimensions with long short-term memory (left) and bidirectional long short-term memory (right) model.

types, utilizing OpenSMILE features with 88 dimensions. Figs. 10 and 11 yield comparable results for OpenSMILE features with 25 dimensions and 88 dimensions, respectively. Although the misclassification rate for all attack types (excluding A17, A18, and A19) remains below 7% for both features with LSTM and BiLSTM, challenges arise in accurately detecting attacks A17 and A18.

Notably, all models with CQCC, MFCC, and OpenSMILE features succeeded in detecting the attack types but exhibited higher rates of misclassification for A17 and A18. The BiLSTM model had the lowest misclassification rate for attack A17 of 11.66% with 13 features of MFCC. Subsequently, the BiLSTM model with 13 features of CQCC obtained 45.62% with the lowest misclassification rate among all other types of features. The findings from Figs. 6-11 lead to the conclusion that detecting attack types A17, A18, and A19 poses greater difficulty compared to other attack types. While the BiLSTM model excels in detecting attacks A17, A18, and A19, the LSTM model achieves superior results in the term of EER and min-tDCF.

VII. Conclusion

In this paper, an extensive investigation has been conducted on the effect of LSTM and BiLSTM model of the ASVspoof 2019 logical access dataset with a time series-based feature. The use of single and fusion versions of features on unseen attacks affects the spoof detection model. The investigations lead to the conclusion that having a higher number feature of time steps cannot guarantee improvement in the model's performance. In addition, the BiLSTM model outperforms the LSTM almost in all types of features. This indicates the usefulness of the BiLSTM model for time series-based features in contributing the spoof detection. Furthermore, within the logical access subset, the attacks A17, A18, and A19 are more challenging to detect. However, the CQCC feature achieved the lowest EER of 6.15% as a single system and an EER of 4.06% of the fusion model and the highest accuracy among all other features with 93.05% gained with the MFCC feature as a single system.

VIII. Acknowledgment

The work was supported by Soran University. We appreciate the Faculty of Science at Soran University for their financial support, access to resources, and facilitation of the completion of this research.

References

Abdul, Z.K., and Al-Talabani, A.K., 2022. Mel frequency cepstral coefficient and its applications: A review. *IEEE Access*, 10, pp. 122136-122158.

Adiban, M., Sameti, H., and Shehnepoor, S., 2020. Replay spoofing countermeasure using autoencoder and siamese networks on ASVspoof 2019 challenge. *Computer Speech and Language*, 64, pp. 1-10.

Ahmed, N., Khan, J., Sheta, N., Tarek, R., Zualkernan, I., and Aloul, F., 2022. Detecting Replay Attack on Voice-Controlled Systems using Small Neural Networks. In: *2022 IEEE 7th Forum on Research and Technologies for Society and Industry Innovation, RTSI 2022*, pp.50-54.

Bai, Z., and Zhang, X.L., 2021. Speaker recognition based on deep learning: An overview. *Neural Networks*, 140, pp. 65-99.

Chakravarty, N., and Dua, M., 2023. Data augmentation and hybrid feature amalgamation to detect audio deep fake attacks. *Physica Scripta*, 98(9), p. 096001.

Dave, N., 2013. Feature extraction methods LPC, PLP and MFCC in speech recognition. *International Journal for Advance Research in Engineering and Technology*, 1(6), pp. 1-5.

Devesh, K., Pavan, K.V., Ayush, A., and Mahadeva Prasanna, S.R., 2022. *Fake Speech Detection Using OpenSMILE Features*. Springer International Publishing, Berlin.

Eyben, F., Scherer, K.R., Schuller, B.W., Sundberg, J., Andre, E., Busso, C., Devillers, L.Y., Epps, J., Laukka, P., Narayanan, S.S., and Truong, K.P., 2016. The Geneva minimalistic acoustic parameter set (GeMAPS) for voice research and affective computing. *IEEE Transactions on Affective Computing*, 7(2), pp. 190-202.

Eyben, F., Wöllmer, M., and Schuller, B., 2010. OpenSMILE - The Munich Versatile and Fast Open-Source Audio Feature Extractor. In: *MM'10-Proceedings of the ACM Multimedia 2010 International Conference*, pp.1459-1462.

Hassan, F., and Javed, A., 2021. Voice Spoofing Countermeasure for Synthetic Speech Detection. In: *2021 International Conference on Artificial Intelligence, ICAI 2021*, pp. 209-212.

Hochreiter, S., and Schmidhuber, J., 1997. Long short-term memory. *Neural*

Computation, 9(8), pp. 1735-1780.

Jiang, Z., Huang, H., Yang, S., Lu, S., and Hao, Z., 2009. Acoustic Feature Comparison of MFCC and CZT-Based Cepstrum for Speech Recognition. In: *5th International Conference on Natural Computation, ICNC 2009*, 1(200808003), pp.55-59.

Kamble, M.R., Sailor, H.B., Patil, H.A., and Li, H., 2020. Advances in anti-spoofing: From the perspective of ASVspoof challenges. *APSIPA Transactions on Signal and Information Processing*, 9, e2.

Karo, M., Yeredor, A., and Lapidot, I., 2024. Compact time-domain representation for logical access spoofed audio. IEEE/ACM Transactions on Audio Speech and Language Processing, 32, pp.946-958.

Kinnunen, T., Delgado, H., Evans, N., Lee, K.A., Vestman, V., Nautsch, A., Todisco, M., Wang, X., Sahidullah, M., Yamagishi, J., and Reynolds, D.A., 2020. Tandem assessment of spoofing countermeasures and automatic speaker verification: Fundamentals. *IEEE/ACM Transactions on Audio Speech and Language Processing*, 28, pp. 2195-2210.

Kinnunen, T., Sahidullah, M., Delgado, H., Todisco, M., Evans, N., Yamagishi, J., and Lee, K.A., 2017. The ASVspoof 2017 Challenge: Assessing the Limits of Replay Spoofing Attack Detection. In: Proceedings of the Annual Conference of the International Speech Communication Association, Interspeech, 2017-August, pp.2-6.

Kumari, T.R.J., and Jayanna, H.S., 2015. Comparison of LPCC and MFCC Features and GMM and GMM-UBM Modeling for Limited Data Speaker Verification. In: *2014 IEEE International Conference on Computational Intelligence and Computing Research, IEEE ICCIC 2014*, pp. 95-103.

McFee, B., Raffel, C., Liang, D., Ellis, D.P.W., McVicar, M., Battenberg, E., and Nietok, O., 2015. Librosa: Audio and Music Signal Analysis in Python. In: *Proceedings of the 14th Python in Science Conference*, (Scipy), pp.18-24.

Nautsch, A., Wang, X., Evans, N., Kinnunen, T., Vestman, V., Todisco, M., Delgado, H., Sahidullah, M., Yamagishi, J., and Lee, K.A., 2021. ASVspoof 2019: Spoofing countermeasures for the detection of synthesized, converted and replayed speech. *IEEE Transactions on Biometrics, Behavior, and Identity Science*, 3(2), pp. 252-265.

Novoselov, S., Kozlov, A., Lavrentyeva, G., Simonchik, K., and Shchemelinin, V., 2016. STC Anti-Spoofing Systems for the ASVspoof 2015 Challenge. In: *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, pp.5475-5479.

Patel, T.B., and Patil, H.A., 2015. Combining Evidences from Mel Cepstral, Cochlear Filter Cepstral and Instantaneous Frequency Features for Detection of Natural vs. Spoofed Speech. In: *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, pp.2062-2066.

Rahmeni, R., Aicha, A.B., and Ayed, Y.B., 2020. Acoustic features exploration and examination for voice spoofing counter measures with boosting machine learning techniques. *Procedia Computer Science*, 176, pp. 1073-1082.

Siami-Namini, S., Tavakoli, N., and Namin, A.S., 2019. The Performance of LSTM and BiLSTM in Forecasting Time Series. In: *Proceedings - 2019 IEEE International Conference on Big Data, Big Data 2019*, pp.3285-3292.

Tian, X., Xiao, X., Chng, E.S., and Li, H., 2017. Spoofing Speech Detection using Temporal Convolutional Neural Network. In: *2016 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference, APSIPA 2016*.

Todisco, M., Delgado, H., and Evans, N., 2016. A New Feature for Automatic Speaker Verification Anti-Spoofing: Constant Q Cepstral Coefficients. In: *Odyssey 2016: Speaker and Language Recognition Workshop*, pp.283-290.

Todisco, M., Delgado, H., and Evans, N., 2017. Constant Q cepstral coefficients: A spoofing countermeasure for automatic speaker verification. *Computer Speech and Language*, 45, pp. 516-535.

Todisco, M., Wang, X., Vestman, V., Sahidullah, M., Delgado, H., Nautsch, A., Yamagishi, J., Evans, N., Kinnunen, T., and Aik Lee, K., 2019. ASVSpoof 2019: Future Horizons in Spoofed and Fake Audio Detection. In: *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, 2019, pp.1008-1012.

Wang, X., Yamagishi, J., Todisco, M., Delgado, H., Nautsch, A., Evans, N., Sahidullah, M., Vestman, V., Kinnunen, T., Lee, K.A., Juvela, L., Alku, P., Peng, Y.H., Hwang, H.T., &… Ling, Z.H., 2020. ASVspoof 2019: A large-scale public database of synthetized, converted and replayed speech. *Computer Speech and Language*, 64, 101114.

Wei, C., Pang, R., and Kuo, C.C.J., 2024. A Green Learning Approach to Spoofed Speech Detection. In: *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing* (*ICASSP*), pp.12956-12960.

Wu, Z., Kinnunen, T., Evans, N., Yamagishi, J., Hanilci, C., Sahidullah, M., and Sizov, A., 2015. ASVspoof 2015: The First Automatic Speaker Verification Spoofing and Countermeasures Challenge. In: *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, pp.2037-2041.

Wu, Z., Yamagishi, J., Kinnunen, T., Hanilçi, C., Sahidullah, M., Sizov, A., Evans, N., Todisco, M., and Delgado, H., 2017. ASVspoof: The automatic speaker verification spoofing and countermeasures challenge. *IEEE Journal on Selected Topics in Signal Processing*, 11(4), pp. 588-604.

Yang, J., Das, R.K., and Li, H., 2020. Significance of subband features for synthetic speech detection. *IEEE Transactions on Information Forensics and Security*, 15(c), pp. 2160-2170.

Zhou, J., Hai, T., Jawawi, D.N.A., Wang, D., Ibeke, E., and Biamba, C., 2022. Voice spoofing countermeasure for voice replay attacks using deep learning. *Journal of Cloud Computing*, 11(1), 51.