IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 31 May 2023, accepted 12 June 2023, date of publication 16 June 2023, date of current version 19 July 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3286864

RESEARCH ARTICLE

Detecting Fake Audio of Arabic Speakers Using Self-Supervised Deep Learning

ZAYNAB M. ALMUTAIRI[1] **AND HEBAH ELGIBREEN**[1,2]

[1]Information Technology Department, College of Computer and Information Sciences, King Saud University, Riyadh 11451, Saudi Arabia
[2]Artificial Intelligence Center of Advanced Studies (Thakaa), King Saud University, Riyadh 11451, Saudi Arabia

Corresponding author: Zaynab M. Almutairi (442202923@student.ksu.edu.sa)

ABSTRACT One of the most significant discussions in forensics is Audio Deepfake, where AI-generated tools are used to clone audio content of people's voices. Although it was intended to improve people's lives, attackers utilized it maliciously, compromising the public's safety. Thus, Machine Learning (ML) and Deep Learning (DL) methods have been developed to detect imitated or synthetically faked voices. However, the developed methods suffered from massive training data or excessive pre-processing. To the author's best knowledge, Arabic speech has not yet been explored with synthetic fake audio, and it is very limited to the challenged fakeness, which is imitation. This paper proposed a new Audio Deepfake detection method called Arabic-AD based on self-supervised learning techniques to detect both synthetic and imitated voices. Additionally, it contributed to the literature by creating the first synthetic dataset of a single speaker who perfectly speaks Modern Standard Arabic (MSA). Besides, the accent was also considered by collecting Arabic recordings from non-Arabic speakers to evaluate the robustness of Arabic-AD. Three extensive experiments were conducted to measure the proposed method and compare it to well-known benchmarks in the literature. As a result, Arabic-AD outperformed other state-of-the-art methods with the lowest EER rate (0.027%), and high detection accuracy (97%) while avoiding the need for excessive training.

INDEX TERMS Audio deepfake, imitation fakeness, Arabic-AD method, modern standard Arabic (MSA), machine learning (ML), deep learning (DL).

I. INTRODUCTION

The recent growth of AI-synthesized tools has demonstrated their power in generating convincing voices [1], which leads to the spread of disinformation using audio around the world [2]. Even though these tools were introduced to improve people's lives, as in creating audiobooks [3], their malicious use caused the fear of a technology known as Audio Deepfake (AD). AD is a new technology that allows users to create audio clips of people saying things they did not say [2]. A recent survey defined AD technology as a speech that has been produced synthetically or modified to sound real [4]. AD was initially developed to improve human life in a variety of applications. For example, assisting people who have lost their voices due to a throat disease or other medical concerns [5], [6] and simulating calming voices in audiobooks [3]. However, attackers used AD technology for malicious intents, where a new AI scam cloned a teenage girl's voice to call her mother and demand a $1 million ransom [7]. It was used to target not only individuals but also banks and companies. In 2020, bank robbers steal $35 million by synthesizing the boss's voice [8]. Moreover, in 2019, criminals impersonated a CEO's audio using AI-based software and swindled more than $243,000 while calling over the phone [9]. Furthermore, politicians and governments may also be affected by AD technology danger [10]. As a result, people all around the globe are starting to worry about the impact this technology may have on their data and the security of their businesses and governments. That is why it is crucial to verify the authenticity of any audio files before they are made public; otherwise, they might be used to propagate false information. Thus, this problem has been an interest of the research community in recent years. In the AD area, different types of fakeness have emerged, such as imitation-based and

The associate editor coordinating the review of this manuscript and approving it for publication was Wai-Keung Fung.

synthetic-based, which increased the challenge in detection. Imitation-based fake audio generation methods date back to 2012, when Ballesteros and Moreno [11], proposed them as a means of securely transmitting confidential information. There are two ways to imitate speech, masking algorithms, such as Efficient Wavelet Mask (EWM) [12], and the traditional way is using humans who have similar voices. On the other hand, synthetic-based, also known as Text-To-Speech (TTS) methods, aimed to transform the text into acceptable and natural speech [13].

To detect AD, many ML and DL methods have been published in the literature, and new models have been constructed. These include supervised ML methods like the Support Vector Machine (SVM) and DL algorithms like the Convolutional Neural Network (CNN). Although these methods performed well in detecting AD, current ML methods have suffered from excessive training and pre-processing, while DL algorithms need special data transformation processes to perform well, not to mention the absence of Arabic AD detection methods. More specifically, there was a trade-off between accuracy and computing complexity. When ML algorithms were first created, their accuracy was relatively good, but they needed excessive training and data pre-processing. When DL algorithms were created, on the other hand, special transformation was required on audio files to be managed by the algorithm. When it comes to Arabic AD, literature is lacking. Only one study [14] was found where the imitation-based fakeness of Classical Arabic (CA) was detected using classical ML and DL methods. There does not seem to be any study that investigates both synthetic and imitation fakeness in Arabic, to the author's knowledge. In addition, investigating accents factor is currently absent from the AD literature, and it is unclear if it influences the accuracy of AD detection methods. To increase the performance of AD detection methods and close the observed gaps in the literature, more research is necessary.

Recently, Self-Supervised DL (SSL) methods have shown promising results in speech recognition and classification, where they can handle raw audio data. The main goal of the SSL is to discover general representations from extensive data without requiring human annotations, which is time-consuming mission [15]. After seeing huge success in computer vision and Natural Language Processing (NLP), SSL was recently embraced for use in voice processing [15]. This technique deals with the dynamic structure that leads to generalization ability and has been inspired by the cognitive process of producing general representation [15]. Consequently, the SSL methods alleviate the issues of classical ML and DL methods. However, from the literature, it was found that SSL has not been utilized for Arabic AD detection methods and the gaps of ML and DL methods are still exiting. Additionally, fake Arabic datasets are still limited especially with synthetic-based fakeness. Thus, this paper contributes to the literature by proposing a new methodology to detect AD from different sources of data. Particularly, a new synthetic audio dataset is built to contribute to the body of the literature and create the first synthetic dataset of MSA single speaker for AD detection. Moreover, two type of Arabic datasets is also collected to train the newly developed AD detection models; including dataset for (1) CA audio speakers with imitation-based fakeness, and (2) MSA synthetic-based multi-speakers called the Arabic-CAPT dataset [16]. In addition to the datasets, this paper contributes to the literature by developing a new SSL AD detection method that results with three finetuned models. The developed method is used to detect fake audio of Arabic speech in imitation and synthetic fakeness, in addition to speech given by multi-speakers from different nationalities with accents. The models resulting from the proposed method are also evaluated and benchmarked with well-known methods in the literature, to measure their accuracy and robustness. In the first experiment, the method efficiency against imitation fakeness was assessed by testing the first model's performance that was fine-tuned over the imitation fakeness dataset. In the second experiment, the method effectiveness against synthetic fakeness is evaluated by measuring accuracy of the second model that was fine-tuned over the single speaker dataset. Moreover, the robustness of the proposed method is examined in the third experiment, by comparing its sensitivity across varied accents of multi-speakers using the third model that was fine-tuned over the collected multi-speakers synthetic-based fakeness dataset.

II. CONTRIBUTION

We can summarize our contribution as follows:

- Create a new single-speaker synthetic dataset for the Arabic MSA language.
- Develop a novel SSL method, called Arabic-AD, for detecting AD of Arabic speech and train new models with different types of fakeness and accents.
- Conduct different evaluation experiments to benchmark the proposed method with well-known methods in the literature and different AD fakeness types, to measure how well various levels of fakeness and accents work with the proposed SSL method.

This article is organized as follows: Section III is the literature review. Section IV presents the contribution of this paper, including the developed AD method and the dataset generation methodology. Section V presents the experimental evaluation and highlights important findings. Section VI concludes the article with future work recommendations.

III. RELATED WORK

In 2022, we have conducted a review paper [17] discussing the related literature covering classical supervised ML and DL methods in AD area, in addition to the available AD detection datasets. Based on the review results, we found that most ML methods are more accurate than DL methods, especially when the speech is in Arabic, while in English, DL methods are more accurate than ML, as confirmed in [18]. It is worth noting that when considering the huge number of audio files, the scalability of ML methods is not proved because of the extensive training and human

pre-processing required. Additionally, although DL methods avoid manual feature extraction and excessive training, they still require special transformations for audio data. Hua et al. [19] proposed a novel model named Time-domain Synthetic Speech Detection Net (TSSDNet) based on two CNN architectures: ResNet and Inception. The main goal of this model is to use the lightweight end-to-end Deep Neural Network (DNN) detection style without using hand-crafted features such as the constant Q transform (CQT) and Constant Q Cepstral Coefficients (CQCC) for better performance. The proposed model is better than state-of-the-art hand-crafted models in detection by 1.96% of EER tested over cross-dataset (Automatic Speaker verification (ASV) spoof 2019 [22] and ASV spoof 2015 challenges [20]) However, the proposed model performance was slightly decreased while applied on a cross-dataset. Thus, SSL DL methods have recently been introduced into the AD detection literature.

A new segmenting detection strategy has been proposed by Xiao et al. [21] for identifying partially synthetic audio, which depends on using a Transformer Encoder (TE). This strategy mainly focused on splitting the audio input signals into segments as the first step, then transforming them into the detection model. Further, the segmented audio input feature will be extracted by the Short-Time Fourier Transform (STFT) process and fed directly to the transformer layer for capturing the anomaly information. The classifier then makes the ultimate determination as to whether the speech is fake. However, the proposed detection strategy did not have an ideal performance due to achieving an Equal Error Rate (EER) score of 40.50% over ADD2022 dataset [22]. Zhang et al. [23] proposed a new detection scheme that depends on the TE with the ResNet network (TE-ResNet). In this research, the efficacy of the proposed method was improved via the use of five augmentation strategies, which were used across ASV spoof 2019 [24] and Fake or Real (FOR) [25]. datasets. In addition, log power spectrum (LPS), Mel Frequency Cepstral Coefficients (MFCC), and CQCC were retrieved from the input as front-end acoustic characteristics for efficient operation. To clarify, ResNet is utilized to compute false detection scores, while a TE is employed to analyze front-end acoustic information to extract from them deep feature maps. The experimental results showed that the proposed method outperformed other existing detection methods with 3.99% on ASV data while achieving 5.89% on FOR data measured by the EER metric. However, when used in a cross-dataset experiment, TE-ResNet fails to perform reliably.

To improve the literature on SSL DL, pre-trained feature extraction models have been adopted into the architecture of DL methods. M. Martın-Donas and Alvarez [26]. proposed a method that relied on the wav2vec 2.0 pre-trained feature extractor and a downstream classifier. The primary motivation for using wav2vec 2.0 was the ability to extract discriminating data efficiently and accurately for use in detecting faked audio. In this work, different data augmentation processes have been adapted to be compatible with the classifier layer. The proposed model was then tested on two datasets (the ASV spoof 2019 [24] and the ADD challenge [22]), where it attained a detection rate of EER 4.98 via experimentation. However, the proposed method used excessive data augmentation processes. In addition, Tak et al. [27] examined several front-end architecture dependent on data augmentation methodologies using the same proposed method. The first front-end architecture was wav2vec 2.0, while the second was Sinc-layer. The experimental results showed that the proposed method produced EER of 2.85% and 20.04% by wav2vec 2.0 and Sinc-layer, respectively. Despite wav2vec 2.0's low EER being preferable to that of the Sinc-layer model, its complicated structure necessitates a larger computing resource. Xie et al. [28] proposed a novel AD detection method based on a Siamese network and wav2vec feature extractor through two phases of learning representation. The first phase focused on learning the features extracted from wav2vec. The second step included training a classifier using the Siamese network's embedding. To accomplish the classification objective, the Siamese network integrates Light CNN, SENet, and ResNet, three traditional DL models, into a Multilayer Perceptron (MLP) final layer. This means that the proposed method has an extremely low EER of just 1.21% when classifying fake audio from real ones. However, the classifier does not contain Batch Normalization (BN) in the network output, which is expected to produce a high generalization error.[1] Likewise, Cai et al. [30] proposed a new detection method based on frame-level boundary detection for detecting partially synthetic audio from real ones. The proposed method starts by extracting features using the wav2vec 2.0-base pre-trained model and then using 1D-ResNet to obtain frame-level embeddings from the extracted features. After that, the embeddings were fed directly to the TE block, which contains Bidirectional Long Short-Term Memory (BiLSTM) for making the embedding in sequential order, followed by a Fully Connected (FC) layer in the end for prediction. As a result, the proposed method provides a detection rate of EER 6.58%. Yet, this method presents a complex structure and needs more improvement to gain good performance with the lowest EER. Liu et al. [31] proposed a novel detection strategy called greedy fusion to detect low-quality fake audio from the ADD challenge dataset [22]. The authors also compared the performance of different SSL-based pre-trained models in detecting partially faked audio from real ones. For the greedy fusion method, which employs a combination of the three conventional models SE-Res2Net50, RawNet2, and ResNet- Temporal Convolutional Network (TCN) for detection, the data required augmentation before feeding it to the network. In the case of the comparison of the three SSL-based methods (wav2vec 2.0-large, WavLM-large, and XLSR), no data augmentation processes were needed, and the detection network consisted of wav2vec 2.0-large, WavLM-large, and XLSR. The greedy fusion method was

[1] Generalization means how well the learned representations captured the ''similarity'' between various entities to reduce loss function and improve performance [29].

not robust since it provided the highest EER of 25.91% when tested under a noisy environment, while XLSR obtained a better EER of 20.58% when compared to wav2vec 2.0-large and WavLM-large. Wang et al. [32] proposed a fully automated end-to-end fake audio detection method based on a wav2vec 1.0 pre-trained model as a feature extractor. In particular, the authors proposed a novel classifier called light-DARTS and were inspired by the previous version of Differentiable Architecture Search (DARTS). The primary motivation for creating this method was to eliminate the need for manual training of deep representations of speech. The results showed that the proposed method delivers a significant improvement in the detection rate by decreasing EER rate of 1.08 over the prior literature. Although light-DARTS achieved the lowest EER in detection, it was not robust and slightly overlapped the classes in the boundary detection during the classification task.

Based on SSL DL methods reviewed so far, it is possible to infer that, although lowering the amount of effort required to handle audio data, AD detection methods that rely mostly on SSL models introduced a high error detection rate. Future research could thus focus on taking advantage of SSL learning while improving its performance. When it comes to the newly published AD detection datasets, a new dataset was generated by eleven speech generation techniques called Fake Audio Detection (FAD) [33]. This dataset consists of 1024 real voices and 279 fake ones also PF samples were considered. Recently, a new MSA dataset for non-native Arabic speakers has been developed for a speech recognition task called Non-Native Arabic Speech Corpus (Arabic-CAPT) [16]. This dataset consists of 63 non-native Arabic speakers from 20 different nationalities. With a file length of 10s (seconds), each speaker clearly mispronounces MSA. Besides, Fastspeech 2 model was the generation technology used to generate this data within 3h (hours) both fake and real samples. However, this dataset was not adapted to AD detection task and was used only in the speech recognition task. Ultimately, although the research has adopted pre-trained feature extraction with SSL DL to overcome traditional ML methods, the proposed methods have not been robust and have been ignored in the examination of other types of fakeness in the AD domain, such as imitation. Moreover, when it comes to AD detection methods for Arabic, only one article targeted the language [13], and it was only for imitation fakeness with the classical ML and DL methods. To the best of the author's knowledge, no prior research has explored Arabic speech based on synthetic fakeness. Furthermore, the most datasets have been developed for non-Arabic languages. Only one dataset used for the AD detection task was found for CA language, but it covered only imitation fakeness. One other synthetic fake audio dataset was discovered in the literature; however, it only featured non-Arabic speakers and was utilized for speech recognition tasks rather than detecting fakes. To the authors' knowledge, no work has investigated the effect of Arabic accents on fake audio detection. The following section will introduce the contribution of this article to overcoming the identified gaps listed before. When it comes to the datasets, imitation and synthetic fakeness datasets will be collected, and a new single-speaker synthetic-based dataset will be created. This will contribute three dataset variations that allow us to test the proposed detection method over single synthetic and imitated fake audio in addition to multi-speaker synthetic accented audio. Moreover, the next section will also introduce a new SSL DL method that can handle both the synthetic and imitation fakeness of Arabic speakers. The main significance of this article is developing a new AD detection method to detect both imitation and synthetic Deepfake with minimal pre-processing while being robust towards different speakers and accents.

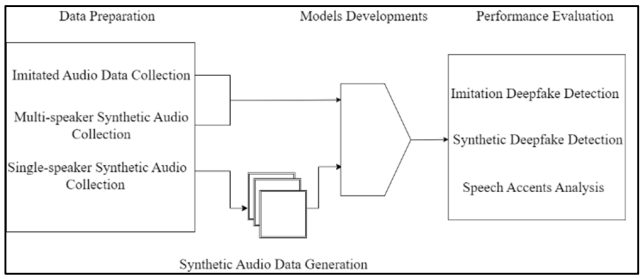

FIGURE 1. The methodology structure.

IV. PROPOSED METHODOLOGY

The proposed research methodology, illustrated in Fig.1, starts by collecting and building the required data for imitation and synthetic Deepfake detection. Consequently, a new AD detection method is proposed, and three different models are developed and fine-tuned. Finally, the developed models are evaluated in different setups and compared to other benchmarks in the available literature.

A. DATA PREPARATION

To be able to create a robust AD detection method, three datasets must be collected and prepared. The first dataset is for Arabic speaker's audio with imitation-based fakeness. The second dataset is for Arabic speaker audio with synthetic-based fakeness. Finally, the third dataset is for multi-speakers with accents, where the speakers are non-Arabic but speak Arabic. During the compilation of the required datasets, it was discovered that certain speech data is accessible in the literature and must be gathered and prepared, while other datasets only include real audio and fake audio needs to be generated. Consequently, in this part, we will describe the collection and pre-processing steps of the datasets that were gathered in addition to the methods applied to build the missing datasets. These ready-to-use datasets will later be utilized to construct the proposed method.

B. IMITATED AUDIO DATA COLLECTION

The first step of the proposed methodology is to collect an Arabic imitation dataset. The collected dataset is called Arabic Diversified Audio (Ar-DAD) [34], which imitated using the traditional way (by humans who have similar voices

TABLE 1. The summary of the Ar-DAD dataset recordings information.

Reciter Name	Reciter Dialect	#Real files	#Imitated files
AbdulBasit AbdulSamad	Egypt	527	28
Ahmed Neana		527	-
Akram Al-Alaqimy		527	-
Ali Hajjaj AlSuesy		527	-
Mahmoud Al-Husary		527	26
Mohammad Al-Minshawy		527	13
Mohammad Al-Tablawy		527	-
Mohammad Jibreel		527	-
Yaser Salamah		527	-
Abdullah Al-Juhaynee	KSA	527	8
Abdullah Basfar		527	7
Abdullah Matroud		527	-
Abdurrahman As-Sudais		527	38
Abu Bakr Ashaatree		527	43
Ahmed Al-Ajamy		527	16
Ali Jaber		527	8
Saad Al-Ghamadi		527	11
Hani Al-Rifai		527	-
Ali Al-Hudhaify		527	33
Maher Al-Muaiqly		527	51
Mohammad Ayyoub		527	22
Nasser AlQatami		527	2
Sahl Yassin		527	-
Salah Al-Budair		527	11
Saood Ashuraym		527	39
Yasser Ad-Dussary		527	5
Meshari Al-Afaasy	Kuwait	527	30
Fares Abbad	Yemen	527	6
Mohammad AbdulKareem	Sudan	527	-
Salah Bukhatir	UAE	527	-
Total		**15,810**	**397**

(-) means that imitation audio for this reciter was not available.

to the target persons). The main reason for collecting Ar-DAD dataset is because of contains both real and imitated voices of Quran reciters. This dataset consists of 30 male Arabic reciters and 12 imitators collected from YouTube channels, the Holy Quran audio portal [35], and websites. Specifically, the recordings of reciters were collected from the Holy Quran audio portal [31] and YouTube channels, while the recordings of imitators were collected from websites. Furthermore, both reciters and imitators are from Saudi Arabia, Kuwait, Egypt, Yemen, Sudan, and the United Arab Emirates (UAE). As illustrated in Table 1, Ar-DAD contains 397 imitated audios that were chosen based on how good the imitators are at imitating the reciters and 15,810 real audio files. In this research, the whole Ar-DAD dataset is used, where in total, 16,207 files containing real and imitated ones. Each file in this dataset is 10s long and stored in WAV format.

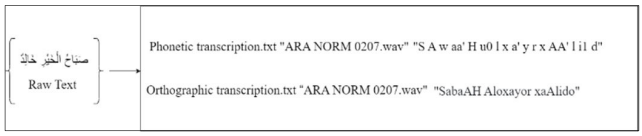

FIGURE 2. The difference between phonetic and orthographic transcription in the ASC dataset.

C. SINGLE SPEAKER SYNTHETIC AUDIO COLLECTION

To train a model that can detect the synthetic fakeness of a single speaker, it also should be fine-tuned using synthetically faked voices. Although different author has explored synthetic dataset, such as in [36] but this effort was focusing on align speech recording (real audio) with its phonetic transcription to be used in generating synthesizing faked audio. Thus, there is no Arabic dataset for synthetic audio of a single speaker that can be used for the proposed method. Consequently, a new Arabic fake audio dataset needs to be generated for this purpose. The generation methodology will be discussed and explained in the next section, but it is important to present the details of the real speaker's audio data that was collected for this purpose. To generate the fake audio speech, real audio from the Arabic Speech Corpus (ASC) [37] was collected. This dataset contains 3h of recorded high-quality MSA language from a male speaker who speaks the language perfectly, where each sample duration lasting between 2s to 30s. What is important in this dataset is that each audio file was also scripted with a phonetic and an orthographic transcript, which is needed later in the generation phase. As shown in Fig.2, the main difference between phonetic and orthographic transcription is that phonetic transcription is a txt file that contains each wave file name followed by the phoneme sequence [37], while orthographic transcription is a txt file that contains each sentence converted into Buckwalter format [37].

ASC also included TextGrids files that can be used later to train the fake audio generator. TextGrid file can be opened by a software called Praat [38] and, as illustrated in Fig. 3, these files define the phoneme labels with time stamps of the WAV files' interval boundaries [37].

One limitation in the ASC dataset is that it does not contain the lexicon files that are necessary for fake audio generation. The lexicon is a txt file that is similar to a dictionary and contains the keys that are extracted from the orthographic transcript, where each key is followed by its phonetic equivalent. In order to generate the lexicon files, we used a library called Arabic-Phonetizer developed by Nawar Halabi [39]. Fig. 4 shows an example of lexicon content, where the key ''>Hmad'' is a word that pronounced by speaker and unique which should not be redundant in the lexicon file, while the phonetic is the sequence of that key.

D. MULTI-SPEAKER SYNTHETIC AUDIO COLLECTION

Research shows that most AD detection methods only look for one form of fakeness, ignoring other factors that might affect their detection accuracy. One such factor is 'accents', which are defined as the way a specific group of people

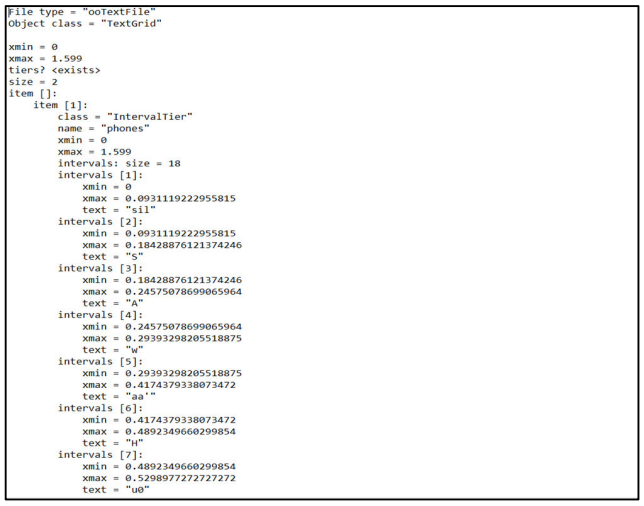

FIGURE 3. An example of textgrid file from the ASC dataset.

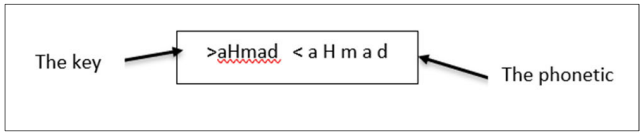

FIGURE 4. An example of lexicon content.

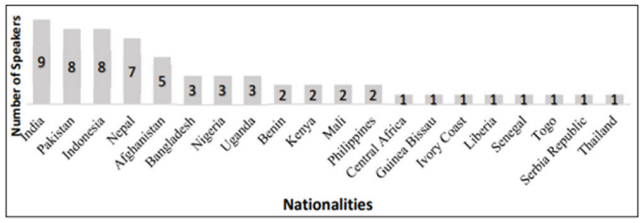

FIGURE 5. The nationalities and the number of speakers for each from Arabic-CAPT dataset [16].

typically speak, particularly the citizens or natives of a particular country [40]. Even though accents have a significant impact on speech recognition models performance [41], there is a lack of studies addressing this issue in the AD detection literature. Thus, it is presently unclear whether accents can affect detection accuracy.

For that reason, it is important to train and evaluate a model using the proposed AD detection method with a multi-speaker dataset that includes accents. However, no datasets that were made for this purpose can be found in the AD detection literature. Recently, a synthetic speech dataset called Arabic-CAPT [16] was published in 2022 and used for mispronunciation detection. This dataset was an extension of the real audio dataset named KSU Arabic Speech Database [42]. In particular, the Arabic-CAPT dataset was collected for this research, containing 3h[2] of real and synthetic MSA speech from 63 male non-Arabic speakers. Each sample duration is between 2s to 10s. The speakers were from

[2]Note that half an hour from the dataset were redundant, thus, in this research, we used only 2h and half.

TABLE 2. The nationalities and the number of speakers for each from Arabic-CAPT dataset [15].

Raw Text	Phonetic Transcription
[هَلْ هَارَ]	[hal ha2ra]
[ضَمِنْتُ شَغَفَكُمْ]	[Damintu shagafakum]

India, Pakistan, Nepal, Afghanistan, Bangladesh, Nigeria, Uganda, Benin, Kenya, Mali, Indonesia, the Philippines, and others, as shown in Fig. 5. The figure also shows the number of speakers from each nationality, showing that the largest number of speakers were from India, Pakistan, Indonesia, and Nepal.

The speakers in the Arabic-CAPT dataset are from different non-Arabic nationalities and, thus, it was noticed that they had accent in the recordings even though the sentences given to the speakers were in MSA. Table 2 shows examples of the sentences given to the speakers from the Arabic-CAPT dataset.

E. SYNTHETIC AUDIO DATA GENERATION

As discussed before, to the authors' knowledge, there is no published Arabic dataset for synthetic audio of a single speaker that has been proposed for AD detection. Thus, the raw audio of the real speakers collected from the ASC Arabic Speech dataset was used to generate synthesized fake audio. The synthetic AD generation model consists of three modules: text analysis, an acoustic, and a vocoder. The text analysis module will first process the incoming text and convert it into linguistic characteristics. Then, the acoustic module extracts the parameters of the target speaker from the dataset depending on the linguistic features generated from the text analysis module. Last, the vocoder will learn to create speech waveforms based on the acoustic feature parameters, and the final audio file will be generated, which includes the synthetic fake audio in a waveform format. To generate the new AD generation model, FastSpeech 2 method - implemented by ARBML group [43]- was used in this paper. In particular, FastSpeech 2 was used to train a synthetic-based fake audio generator over the collected real audio, as illustrated in Fig. 6. The first step to generating the synthetic audio is loading the ASC dataset with all necessary contents, including wave files, an orthographic transcript, and TextGrids. Consequently, an orthographic transcript will be made when the dataset metadata has been generated for each wave file. Moreover, the wave files are pre-processed (normalized) by the model itself. Following data preprocessing, the model begins extracting phoneme-specific embeddings and applies positional encoding to these embeddings before passing them on to the encoder layer. The model then projects variance features such as F0 (the fundamental frequency of real audio [44]) and energy to the phoneme hidden sequence in the variance adaptor layer, to acquire sufficient information during training, as shown in Fig. 7. After that, a filter with a size of 256 and a kernel with a size of 3 with a dropout of size 0.5 were used to

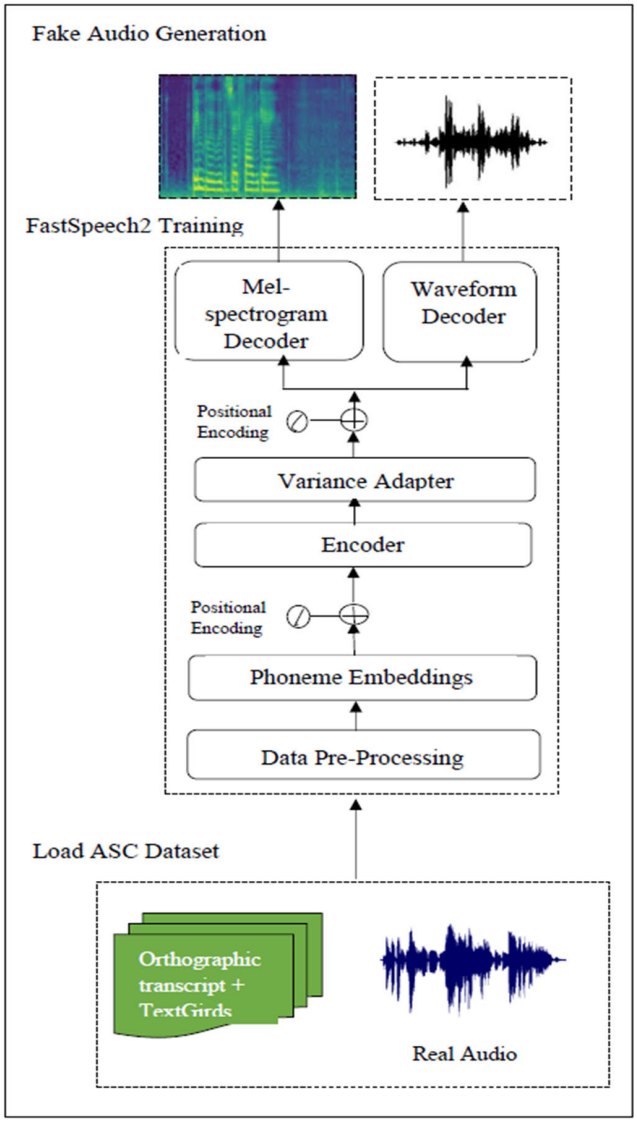

FIGURE 6. Synthetic generation methodology.

avoid overfitting. After the variance adapter layer generates a new hidden sequence, it is necessary to re-encode the prior sequence. Then, in the waveform decoder, we used the HiFi-GAN model with type 'universal' due to its support for multi-languages including Arabic. To optimize the model with the data, Table 3 shows the parameters used by the model. The following link includes samples of the generated fake audio: https://www.kaggle.com/datasets/zaynabalmutairi/arabic-speech-corpus-msa-synthetic-dataset

Once the information has been processed and converted into linguistic features, the FastSpeech 2 model begins training. Then, the acoustic module extracts the parameters of the wave files depending on TextGrids and the linguistic features generated from the text analysis module. Finally, the vocoder learns to create speech waveforms based on the acoustic feature parameters, and the final audio file is generated. The output from the FastSpeech 2 model will include synthetic fake audio in a waveform format. During

TABLE 3. FastSpeech2 parameters.

Parameters	
batch_size: 1	total_step: 250000
multi_speaker: False	anneal_steps: [300000, 400000, 500000]
max_seq_len: 3000	warm_up_step: 4000
save_step: 50000	grad_clip_thresh: 1.0
val_step: 10000	grad_acc_step: 1
synth_step: 1000	weight_decay: 0.0
log_step: 1000	eps: 0.000000001
anneal_rate: 0.3	betas: [0.9, 0.98]

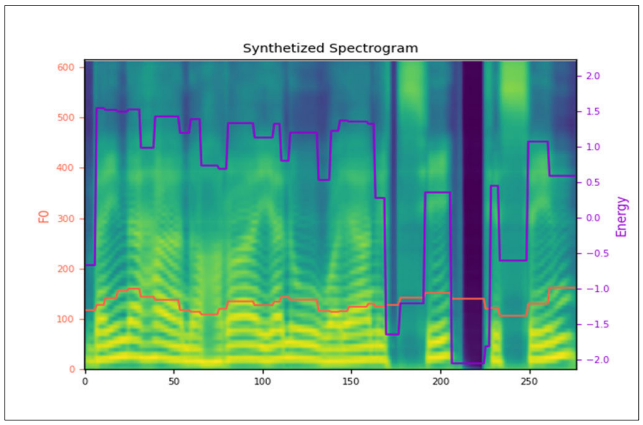

FIGURE 7. An example of the synthetized spectrogram using FastSpeech 2 model.

the training, it was discovered that, when the original audio file is longer than 15s, the resulting fake audio always ends up one second shorter than the original file. This was caused because of adding Attentive Statistical Pooling (ASP) layer to the proposed Arabic-AD method structure. ASP is a layer that calculates the vector of the Mean (μ) with standard deviation vector (σ) for the frame-level features [45]. An existing research paper makes use of this layer to compensate for the difference in audio embedding durations between the two classes (real and fake) [46].

F. AD DETECTION MODELS DEVELOPMENT

Existing research shows that SSL-based models that have already been trained well on a wide range of tasks [47]. Thus, to build classification models that can detect fake audio, a new SSL-based AD detection method is proposed in this paper. In particular, the proposed detection method is inspired by the recent state-of-the-art SSL model called Hidden-Unit Bidirectional Encoder Representations from Transformers (HuBERT) [48]. HuBERT is a pre-trained model that can be adapted to AD detection. It was originally developed to learn acoustic and language models. However, the model's strength is in its ability to learn high-level representations of unmasked inputs taken from audio inputs, and it was designed primarily for use in speech recognition tasks [48].

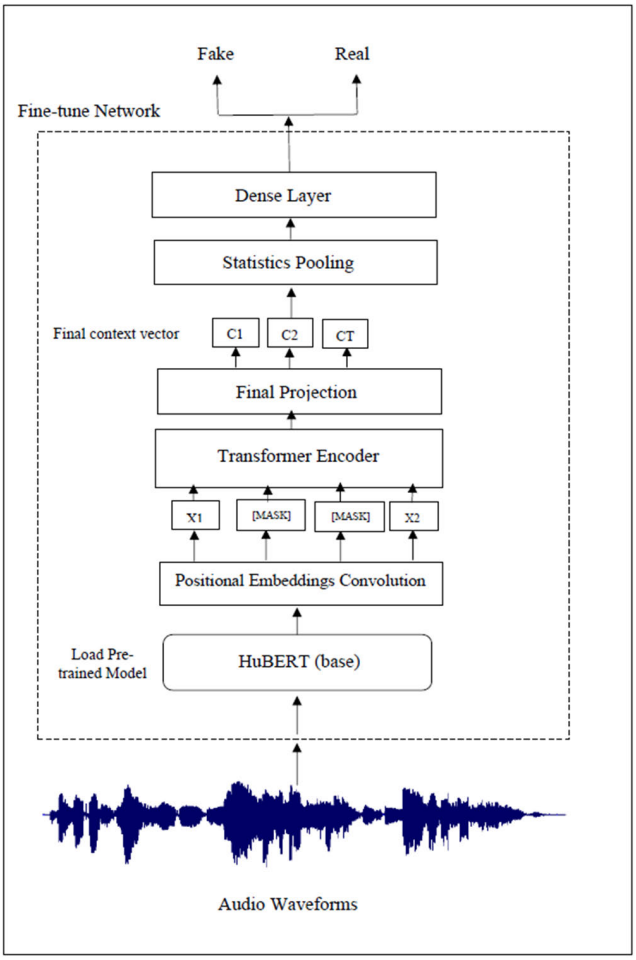

FIGURE 8. The proposed Arabic-AD detection method.

It employs acoustic unit discovery models like Gaussian Mixture Model (GMM) and k-means to provide a frame level for each projected hidden unit [49]. The main reason for choosing HuBERT is because research has demonstrated that large-scale, unlabeled pre-training models can extract useful information [46]. More specifically, we utilize the pre-trained BASE HuBERT model (facebook/hubert-base-ls960) [50], which was trained on unlabeled speech with ∼95M parameters. The main reason for choosing the BASE version of HuBERT is due to its computational limitations; where it needed the fewest resources compared to the other versions. Although HuBERT was used as the basis for the proposed method's pre-trained model, additional layers and blocks have been added to the method architecture to adapt it over the targeted problem and eliminate the need for any excessive data processing. As illustrated in Fig. 8, the proposed method starts by loading the HuBERT pre-trained model and feeding it the audio waveforms. The loaded, pre-trained model will extract the vector representations (embeddings) from the signals of the input audio waveforms directly using a feature convolutional layer. This vector is then partitioned into masked and unmasked audio streams. The pre-trained model then encodes the unmasked inputs into meaningful continuous latent representations. Following that, the encoded, unmasked inputs are fed into the TE to obtain contextualized representations. The resulting contextualized representations are fed into the projection layer to project the final context vector. By doing so, the proposed method will make use of the pre-trained model to better capture the long-term temporal links between the learnt representations and minimize prediction error.

TABLE 4. The proposed Arabic-AD detection method architecture and layers.

Layer Name	Output Size
Conv_dim	512
Hidden_size	768
Projection_Layer_size	256
ASP_Layer_size	256
Num_attention_heads	12
Dense_Layer	128

The resulting context vector from the final projection layer of the pre-trained model is passed after that to the ASP layer with Mean (μ) measure [45]. Lastly, a dense layer with a Tanh activation function is added as the final step of the proposed method. Tanh is an activation function and stands for Hyperbolic Tangent function, which produces zero-centered output for preferred training performance in multi-layer networks [51]. In addition, it works by determining thresholding values between −1 and 1, where the output is defined using eq. (1); where e indicates a mathematical constant which is the base of the natural logarithm, and x indicates the value of the input.

$$f(x) = \left(\frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}\right) - (1.10) \quad [51] \tag{1}$$

The dense layer is used to fine-tune the classification model for the targeted problem. It is also important to note that hyperparameters were tuned using Bayesian Optimizations (BO) [52] during the learning process. In summary, Table 4 shows the proposed method architecture of each layer and its output dimensions.

Based on the method discussed so far, three AD detection models were built and fine-tuned using the collected datasets. Fig. 9 illustrates the training pseudo code, where x = {x1, x2, . . . xn} is a collection of audio recordings spoken by different speakers S; W = {w1, w2, ...wn} is a collection of sentences related to the recordings X, in which each sentence wj equivalent to only one phoneme xj. The main goal of our method is to classify the given waveform audio X and detect if it is fake or real.

This way, it was possible to fine-tune models that take the audio input as WAV files without the need for special transformation or visualization. All audio files across all datasets will need to have their sampling rate lowered to a rate that is compatible with the pre-trained model to work perfectly. Accordingly, the sample rate was reduced from 48000KHZ to 16000KHZ.

Algorithm 1 Training Procedure of Arabic-AD detection Algorithm

Input: Audio Waveform X
Output: Single label (Real or Fake)

```
Procedure Arabic-AD(X) ▷ AD detection
    X ← [] and F ← []
    For i=1 to Max-iter do
        Sample a mini batch of pairs from X
            F ← HuBERT Feature Extractor (Xi)
            Compute M0 as the init. of ASP
            Compute loss for predicted Y
            Print TL
            Print EL
    End for
    Print Xj single label
End Procedure
```

FIGURE 9. The training pseudo code of the Arabic-AD detection algorithm where M indicates the mean, TL indicates training loss and EL indicates evaluation loss.

V. EXPERIMENTAL STUDIES

In order to evaluate the results of the proposed Arabic-AD detection method, several experiments need to be implemented. This section will first present the experimental set up and use of evaluation measures. Then, it will evaluate imitation-based detection. After that, the Arabic-AD detection method's effectiveness against synthetic-based fakeness will be benchmarked. In the third experiment, the accent factor is tested to evaluate the robustness of the Arabic-AD detection method when detecting non-Arabic multi-speakers. At the end, the research questions will be answered, and further literature will be compared with the results.

A. EXPERIMENTAL SETUP

In all experiments, the proposed method was implemented using the 'hugging face' transformer API with Google Collab GPUs. Each experiment involves fine-tuning a unique set of BO hyperparameters to account for the wide variety of data-driven factors. Moreover, a dropout layer is added before the dense layer to avoid overfitting issues. Both CNN and Long Short-Term Memory (LSTM) are used as benchmarks against the built Arabic-AD detection models that are evaluated. The main reason for choosing these two methods and training them over the collected/created datasets is because they have shown better performance in the literature and are usually used for benchmarking. To make the comparison more precise, we experimented with CNN and LSTM with the same parameters used in Arabic-AD in terms of batch size, number of training epochs, and activation function. In all experiments, the macro average[3] is used to evaluate the overall performance of each label and assign equal weights to each class. To measure the performance, different metrics are collected in each experiment. The first metric used is EER, which is used when evaluating the proposed method in comparison to the recent state-of-the-art SSL-based methods. EER is the error rate, where the false-negative (FN) rate and the false-positive (FP) rate are equal [54]. Other measures

[3]Macro average defined as the mean value of a common class-wise metric across all labeled individuals [53].

TABLE 5. The metrics of performance [12].

Metric	Equation
Precision	$P = \frac{TP}{TP + FP}$
Recall	$R = \frac{TP}{TP + FN}$
F1-score	$F1 = 2 \times \frac{1}{\frac{1}{Precision} + \frac{1}{Recall}}$
Overall Accuracy	$OA = \frac{TP + TN}{TP + FN + FP + TN}$

have also been computed in the experiments, as shown in Table 5. F1-score is a score that calculates both FP and FN of the predicted classes, where it depends on the values of precision and recall metrics. Precision is defined as the ratio of correctly anticipated positive results to the total number of positive predictions. The recall, or sensitivity, is the percentage of accurately predicted positive findings to the total number of observations in the actual class. The accuracy is identified as the proficiency measure of the classifier to accurately categorize an object as either normal or attack [55].

In addition, to measure loss in the trained method, the cross-entropy loss function is computed [56]. The loss function calculates the variance between the target label and the output of neural networks. Consequently, fewer values mean better loss, and this function is calculated by the equation (2), where M indicates the number of training examples and ym indicates the target label of the training example m, while *Xm* indicates the inputs of the training example m and *hØ* indicates the method with hidden neural network weights Ø.

$$Jbce = -\frac{1}{M}\sum_{m=1}^{M}\left[ym \times \log\left(h\emptyset\left(Xm\right)\right) + (1 - ym) \times \log\left(1 - h\emptyset\left(Xm\right)\right)\right] \quad [56] \quad (2)$$

Different visualization plots are also introduced in this section, including training/evaluation accuracy and loss curves. The training/evaluation curves visualize the difference in performance between the training and evaluation phases using the accuracy and loss measures per step.

B. IMITATION-BASED DETECTION

In literature, imitation-based investigations have been addressed using classical ML and DL models. To our knowledge, there has been no prior study that looked at this kind of fake using SSL-based methods. Consequently, the pre-trained model is loaded in this experiment, and the proposed Arabic-AD detection method is applied to fine-tune a new model over the Ar-DAD dataset (described in section IV) and test it using a 70:30 split between training and evaluation. The training and evaluation split was made with the 'stratify' parameter to preserve the same label distribution in both samples and avoid a bias split. To preserve a general representation of the input and boost performance, the first two layers of the method are frozen. After that, the BO hyperparameters are tuned, resulting in the values illustrated in Table 6. The goal of using such hyperparameters

TABLE 6. The hyperparameters used in imitation-based detection experiment.

Hyperparameters	
learning_rate=5e-5	per_device_train_batch_size=1
per_device_eval_batch_size=1	num_train_epochs=3.0
weight_decay=0.1	warmup_steps=0.3

is to enhance the efficiency of the method. Weight decay is a regularization hyperparameter used to generalize well for deep neural network training [57]. The learning rate is a tuning parameter in an optimization algorithm that establishes the step size at each iteration as it advances toward the minimization of a loss function [58]. Warmup steps are an effective hyperparameter used for models which exhibit large $\lambda 1$ either in model initialization or at the beginning of training [59]. Batch size is a hyper-parameter that describes the size of the random sample taken from the entire dataset during training [60]. Epochs determines how many times the learning algorithm will run over the whole training dataset [61].

Due to Google Collab GPUs limited memory, the training has taken 1.6K steps during three epochs. In each step, the accuracy and loss were calculated to compare the method's performance during training and evaluation. The following subsections will dive into the detailed results and discuss new findings about the Arabic-AD detection method compared to other better-known methods in the literature.

C. EXPERIMENT RESULTS

As a result of training the Arabic-AD detection method for imitation-based AD detection, it was confirmed that the method achieved high performance while avoiding the need for special data transformation or preprocessing. When comparing the method's accuracy, as illustrated in Fig. 10, the figure shows how the training accuracy increases from 86% to 97% at epoch 2, then remains steady until it peaks at 98%. Similarly, the evaluation accuracy increases from 86% to 96% at epoch 2, then remains steady until it peaks at 97%. As a result, the gap between the training and evaluation indicates that the trained method is not overfitting. In addition to accuracy, it was also important to calculate the loss function during evaluation. As illustrated in Fig. 10, the training and evaluation loss start decreasing in the same direction during the convergence process. In the end, the training and evaluation losses reached 0.18 and 0.14, respectively, indicating a low error rate. The difference between the training and evaluation losses is not large, which also indicates that there is no overfitting issue.

Moreover, CCN and LSTM methods were applied over the collected dataset, and the performance metrics are summarized in Table 7. From Table 7, it can be concluded that the Arabic-AD detection method surpassed the classical DL methods while avoiding the need for special data transformation or preprocessing. Starting with method accuracy, the Arabic-AD detection method achieved significantly higher

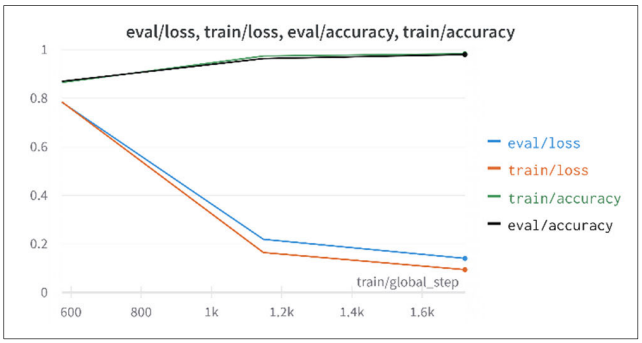

FIGURE 10. Training/evaluation loss curves and eval/training accuracy in imitation-based detection experiment.

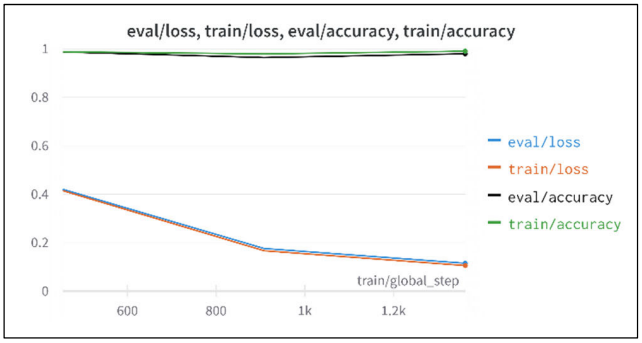

FIGURE 11. Training/evaluation loss curves and eval/training accuracy in synthetic Deepfake detection experiment.

TABLE 7. Comparison between Arabic-AD method results with classical methods in imitation-based detection experiment.

Method	Input	Accuracy	Precision	Recall	F1	EER
CNN	MFCC	71%	61%	61%	61%	0.391%
LSTM	MFCC	90%	95%	81%	85%	0.054%
Arabic-AD	**Wavefo rms**	**97%**	**96%**	**98%**	**97%**	**0.019%**

detection accuracy than LSTM and CNN by approximately 7% and 20%, respectively. However, in precision testing, LSTM achieved a closer rate to Arabic-AD, while CNN reported the lowest rate. Additionally, Arabic-AD outperformed LSTM in terms of Recall and F1-score by 17 and 12, respectively. Compared with CNN, Arabic-AD reported higher scores (by 30%) in both metrics (Recall and F1-score).

When it comes to the EER, it is obvious that Arabic-AD provided the lowest error rate, which supports the loss results discussed before. Although the Arabic-AD detection method deals with inputs directly (waveforms) without the need to transform them into MFCC as image-based methods, it achieved high detection performance in all metrics with the lowest EER. This confirmed that it is an effective method against difficult fakeness, such as imitation.

D. SYNTHETIC DEEPFAKE DETECTION

Effectiveness is one of the performance principles that can be analyzed and described in any ML or DL model. Consequently, in this experiment, we analyze the effectiveness of

TABLE 8. The hyperparameters used in synthetic Deepfake detection experiment.

Hyperparameters	
learning_rate=1e-6	per_device_train_batch_size=1
per_device_eval_batch_size=1	num_train_epochs=3.0
weight_decay=0.1	warmup_steps=0.7

the Arabic-AD detection method while detecting synthetic-based data from a speaker who speaks MSA perfectly. The main reason behind using this data is to test if the Arabic-AD detection method could detect synthetic-based data easily from a speaker who speaks MSA clearly. Consequently, the pre-trained model is loaded in this experiment, and the proposed Arabic-AD detection method is applied to fine-tune a new model over the generated dataset, as explained in section VI. The dataset is divided into 70:30 split between training and evaluation, where the "stratify" parameter is used to preserve the same label distribution in both samples and avoid bias splitting. Following the same procedure as in the previous experiment, the first two layers of the method are frozen to preserve a general representation of the input and boost the performance. After that, the model is fine-tuned using the BO hyperparameters shown in Table 8. Lastly, the final context vector is extracted from the TE layer, and the fine-tuning task in the Dens layer is accomplished.

Due to Google Collab GPUs limited memory, the training has taken 1.2K steps during three epochs. In each step, the accuracy and loss were calculated to compare the method's performance during training and evaluation. The following subsections will dive into the detailed results and discuss new findings about the Arabic-AD detection method compared to other better-known methods in the literature.

E. EXPERIMENT RESULTS

As a result of training the Arabic-AD detection method for synthetic-based AD detection, it was confirmed that the method detects synthetic data effectively. More specifically, when comparing the method accuracy, as illustrated in Fig. 11, the figure shows how the training accuracy remains between 98% and 99% while the evaluation accuracy decreases from 98% to 96% at epoch 2, then increases until it peaks at 97%. The gap between the training and testing evaluations is not far from each other, indicating that the trained method is not overfitting.

In addition to the accuracy, it was also important to calculate the loss function during evaluation. As illustrated in Fig.11, during the convergence phase, the training loss and evaluation loss both start going down and eventually level off at 0.19 and 0.11 on the final epoch. This indicates that the method is not subject to overfitting since there is no significant difference between training loss and evaluation loss. When it comes to the performance of the well-known benchmark, as summarized in Table 9, it can be concluded that the Arabic-AD detection method achieved the same detection accuracy as the LSTM method without the need

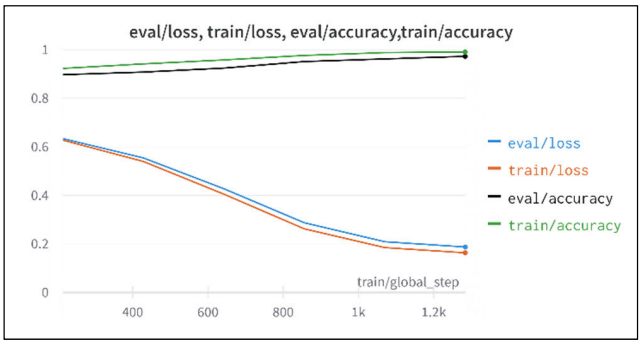

FIGURE 12. Training/evaluation loss curves and eval/training accuracy in speech accents analysis experiment.

TABLE 9. Comparison between proposed method results with classical methods in synthetic Deepfake experiment.

Method	Input	Accuracy	Recall	F1	Precision	EER
CNN	MFCC	93%	94%	93%	92%	0.076%
LSTM	MFCC	97%	98%	98%	97%	0.025%
Arabic-AD	**Waveforms**	**97%**	**98%**	**97%**	**97%**	**0.017 %**

for excessive pre-processing. The CNN method is not an ideal choice in the AD area since it achieved the lowest performance in all metrics compared to Arabic-AD and LSTM. The LSTM method performed similarly to the Arabic-AD detection method, apart from F1-score, where it was 1 percent higher. Despite that, Arabic-AD achieved the lowest EER compared to the others. Since the Arabic-AD did not change one metric for another while outperforming or matching other methods, it is considered as the superior one. It was found that the method achieved results similar to those of classical methods except for EER with avoiding excessive pre-processing. While these numbers may seem high at first glance, they are consistent with what has been discovered in the literature when models are trained using just a single speaker of audio. An example is in the ASR area, where the end-to-end models have received a lot of attention in single speaker with very successful results [62], [63]. Consequently, the following experiment will test the Arabic-AD detection method's robustness by applying it to a diverse set of speakers with varying accents to verify the method's robustness.

F. SPEECH ACCENTS ANALYSIS

This experiment measures how well the Arabic-AD detection method performs when accents are included into the used dataset, providing insight into the method 's robustness. In particular, the Arabic-CAPT dataset that contains multi-speakers of non-Arabic people who do not speak MSA perfectly (explained in section VI) is used to fine-tune a new model through the proposed Arabic-AD detection method. The same procedures configured in the previous experiments are also implemented here. The BO hyperparameters were fine-tuned in this experiment with different values to optimize the performance, as shown in Table 10.

TABLE 10. The hyperparameters used in speech accents analysis experiment.

Hyperparameters	
learning_rate=1e-6	per_device_train_batch_size=1
per_device_eval_batch_size=1	num_train_epochs=6.0
weight_decay=0.4	warmup_steps=0.9

TABLE 11. Comparison between proposed method results with classical methods in speech accents analysis experiment.

Method	Input	Accuracy	Precision	Recall	F1	EER
CNN	MFCC	84%	85%	84%	84%	0.150%
LSTM	MFCC	91%	92%	92%	92%	0.079%
Arabic-AD	**Waveforms**	**97%**	**97%**	**97%**	**97%**	**0.027%**

Due to Google Collab GPUs limited memory, the training has taken 1.2K steps during six epochs. In each step the accuracy and loss were calculated to compare the method performance during training and evaluation. The following subsections will dive into the detailed results and discuss new findings about the Arabic-AD detection method comparing it to other well-known methods in the literature.

G. EXPERIMENT RESULTS

As a result of training the Arabic-AD detection method for multi-speakers using synthetic-based data, it was confirmed that Arabic-AD surpassed the classical DL methods with high detection accuracy and a low EER rate. Fig. 12, depicts the accuracy of the training over time, showing that it continuously improves from 92% in the first epoch to 99% in the final epoch. Similarly, the evaluation accuracy increases steadily from 89 % to the last epoch at 97%. This observation indicates that the method trained well without misbehaving. In addition to accuracy, the loss function is reported in Fig. 12, showing that the training and evaluation losses are decreasing until they reach a stability point of 0.24 and 0.18 respectively, at the end of the training. This observation indicates that the Arabic-AD detection method is optimally fitted.

When comparing to the well-known benchmark, as summarized in Table 11, the Arabic-AD detection method achieved higher detection accuracy than LSTM and CNN by approximately 6 % and 13 %, respectively. Furthermore, in terms of precision, Arabic-AD outperforms LSTM by 5% and CNN by 12%. Correspondingly, Arabic-AD produced good results in Recall and F1-score. When it comes to the EER, it is obvious that Arabic-AD provided the lowest rate, with 0.123 % and 0.052 % reductions compared to CNN and LSTM, respectively.

In general, the performance of the Arabic-AD detection method does not reduce when compared to the previous experiment. This confirms that it is effective and robust against the accent factor of multi-speakers' synthetic data. Even though the Arabic-AD works with inputs directly (waveforms), rather than transforming them into MFCC like the other method does, however, it obtained great detection performance in all measures with the lowest EER. This confirmed that it is an effective and robust method to detect synthetic data of different characteristics, such as accents.

H. DISCUSSION

Once the performance of the Arabic-AD detection method has been verified and compared to base-line classical-based methods, it is also crucial to compare it with the SSL-based already found in the literature. Based on the literature, no studies investigated SSL-based methods yet to detect imitation fakeness and single synthetic-based. However, some studies investigated SSL-based methods to detect synthetic-based fakeness of multi-speakers. When comparing with state-of-the-art SSL-based methods mentioned in the literature, particularly Light-DARTS, Arabic-AD shown significantly low EER. Although the setup and datasets are different, but such significant difference give us an intuition on how the use of pre-trained models (such as Hubert in our case) as an encoder has a significant effect and can minimized the prediction error while capturing the long-term temporal links between the learned representations. Moreover, since Hubert has a clustering process this can overcome the inconsistency problem between targets, which is not considered in the Light-DARTS method. In addition, when Arabic-AD takes accents factor into account, it still reported low EER. According to such findings, Arabic-AD shows high potentials in being more effective in detecting the challenging fakeness types, especially imitation-based fakeness.

VI. CONCLUSION

Detecting fake audio in our societies is becoming more challenging and a crucial issue to tackle. In this paper, a new AD detection method for Arabic speakers was developed, which is called Arabic-AD. Also, a new MSA dataset of single speaker was created synthetically. As a result, Arabic-AD was robust and superior to previous SSL-based methods with a low EER rate while detecting synthetic sounds of speakers that speak MSA differently. Furthermore, it is the first SSL-based method that detects the challenged fakeness type significantly, which is imitation. However, two limitations still exist and need to be addressed in the future, which are the limited availability of fake Arabic datasets and resources. Despite the small size and unbalanced of the datasets employed, the Arabic-AD detection method outperformed benchmark methods. Although our work included 3h of audio datasets, it is better to scale it to more in the future. Thus, in the future, we want to expand the number of datasets while keeping in mind that they should be balanced to assess the performance of Arabic-AD more precisely. Moreover, we will further test Arabic-AD with new SSL methods focusing on multi-speakers' datasets using more resources. Additionally, investigating the Arabic-AD detection method with a new challenging fakeness type, ''replay-attack Deepfake''. Replay-attack fakeness is a type of malicious work that aims to replay a recording of the target speaker's voice.

LIST OF ABBREVIATIONS

AD	Audio Deepfake
Ar-DAD	Arabic Diversified Audio
ASC	Arabic Speech Corpus
ASP	Attentive Statistical Pooling
ASV	Automatic Speaker verification
BN	Batch Normalization
BO	Bayesian Optimizations
BiLSTM	Bidirectional Long Short-Term Memory
CA	Classical Arabic
CQCC	Constant Q Cepstral Coefficients
CQT	Constant Q Transforme
CNN	Convolutional Neural Network
DL	Deep Learning
DARTS	Differentiable Architecture Search
DNN	Deep Neural Network
EWM	Efficient Wavelet Mask
EER	Equal Error Rate
EL	Evaluation Loss
FAD	Fake Audio Detection
HuBERT	Hidden-Unit Bidirectional Encoder Representations from Transformers
Tanh	Hyperbolic Tangent
LPS	log power spectrum
LSTM	Long Short-Term Memory
ML	Machine Learning
MFCC	Mel Frequency Cepstral Coefficients
MSA	Modern Standard Arabic
MLP	Multilayer Perceptron
NLP	Natural Language Processing
Arabic-CAPT	Non-Native Arabic Speech Corpus
SSL	Self-Supervised learning
STFT	Short-Time Fourier Transform
SVM	Support Vector Machine
TCN	Temporal Convolutional Network
TSSDNet	Time-domain Synthetic Speech Detection Net
FoR	Fake or Real
FNR	False-Negative Rate and the
FPR	False-Positive Rate
FC	Fully Connected
GMM	Gaussian Mixture Model

REFERENCES

[1] S. Lyu, "DeepFake detection: Current challenges and next steps," in *Proc. IEEE Int. Conf. Multimedia Expo. Workshops (ICMEW)*, Jul. 2020, pp. 1–6, doi: 10.1109/ICMEW46912.2020.9105991.

[2] N. Diakopoulos and D. Johnson, "Anticipating and addressing the ethical implications of deepfakes in the context of elections," *New Media Soc.*, vol. 23, no. 7, pp. 2072–2098, Jul. 2021, doi: 10.1177/1461444820925811.

[3] A. Chadha, V. Kumar, S. Kashyap, and M. Gupta, "Deepfake: An overview," in *Proc. 2nd Int. Conf. Comput., Commun., Cyber-Secur.*, P. K. Singh, S. T. Wierzchoń, S. Tanwar, M. Ganzha, J. J. P. C. Rodrigues, Eds. Singapore: Springer, 2021, pp. 557–566.

[4] Z. Khanjani, G. Watson, and V. P. Janeja, "Audio deepfakes: A survey," *Frontiers Big Data*, vol. 5, pp. 100–1063, Jan. 2022.

[5] D. Brown. (Aug. 18, 2021). *AI Gave Val Kilmer his Voice Back. But Critics Worry the Technology Could be Misused, The Washington Post*, Accessed: Aug. 21, 2022. [Online]. Available: https://www.washingtonpost.com/technology/2021/08/18/val-kilmer-ai-voice-cloning/

[6] V. Etienne. (Aug. 19, 2021). Val Kilmer Gets His Voice Back After Throat Cancer Battle Using AI Technology: Hear the Results. Peoplemag. Accessed: Aug. 21, 2022. [Online]. Available: https://people.com/movies/val-kilmer-gets-his-voice-back-after-throat-cancer-battle-using-ai-technology-hear-the-results/

[7] J. Smith. (Apr. 11, 2023). *Terrifying New AI Kidnapping Scam Used Teen Girl's Voice to Demand $1m*. Accessed: Apr. 24, 2023. [Online]. Available: https://www.dailymail.co.uk/news/article-11961539/Terrifying-new-AI-scam-used-teen-girls-REAL-voice-call-mother-demand-1million.html

[8] T. Brewster. (2021). Fraudsters Cloned Company Director's Voice in $35 Million Bank Heist, Police Find. Fraudsters Cloned Company Director's Voice In $35 Million Bank Heist, Police Find. Accessed: Apr. 11, 2023. [Online]. Available: https://www.forbes.com/sites/thomasbrewster/2021/10/14/huge-bank-fraud-uses-deep-fake-voice-tech-to-steal-millions/?sh=48216ee07559

[9] C. Stupp. (Aug. 30, 2019). *Fraudsters Used AI to Mimic CEO's Voice in Unusual Cybercrime Case*. The Wall street Journal, Accessed: Jan. 29, 2022. [Online]. Available: https://www.wsj.com/articles/fraudsters-use-ai-to-mimic-ceos-voice-in-unusual-cybercrime-case-11567157402

[10] S. Suwajanakorn, S. M. Seitz, and I. Kemelmacher-Shlizerman, "Synthesizing Obama: Learning lip sync from audio," *ACM Trans. Graph.*, vol. 36, no. 4, pp. 1–13, Aug. 2017.

[11] D. M. Ballesteros L and J. M. Moreno A, "On the ability of adaptation of speech signals and data hiding," *Exp. Syst. Appl.*, vol. 39, no. 16, pp. 12574–12579, Nov. 2012, doi: 10.1016/j.eswa.2012.05.027.

[12] Y. Rodríguez-Ortega, D. M. Ballesteros, and D. Renza, "A machine learning model to detect fake voice," in *Applied Informatics*, H. Florez S. Misra, Eds. Cham, Switzerland: Springer, 2020, pp. 3–13.

[13] X. Tan, T. Qin, F. Soong, and T.-Y. Liu, "A survey on neural speech synthesis," 2021, *arXiv:2106.15561*.

[14] M. Lataifeh, A. Elnagar, I. Shahin, and A. B. Nassif, "Arabic audio clips: Identification and discrimination of authentic cantillations from imitations," *Neurocomputing*, vol. 418, pp. 162–177, Dec. 2020, doi: 10.1016/j.neucom.2020.07.099.

[15] S. Liu, A. Mallol-Ragolta, E. Parada-Cabeleiro, K. Qian, X. Jing, A. Kathan, B. Hu, and B. W. Schuller, "Audio self-supervised learning: A survey," 2022, *arXiv:2203.01205*.

[16] M. Algabri, H. Mathkour, M. Alsulaiman, and M. A. Bencherif, "Mispronunciation detection and diagnosis with articulatory-level feedback generation for non-native Arabic speech," *Mathematics*, vol. 10, no. 15, p. 2727, 2022, doi: 10.3390/math10152727.

[17] Z. Almutairi and H. Elgibreen, "A review of modern audio deepfake detection methods: Challenges and future directions," *Algorithms*, vol. 15, no. 5, p. 155, 2022, doi: 10.3390/a15050155.

[18] J. Khochare, C. Joshi, B. Yenarkar, S. Suratkar, and F. Kazi, "A deep learning framework for audio deepfake detection," *Arabian J. Sci. Eng.*, vol. 47, no. 3, pp. 3447–3458, Nov. 2021, doi: 10.1007/s13369-021-06297-w.

[19] G. Hua, A. B. J. Teoh, and H. Zhang, "Towards end-to-end synthetic speech detection," *IEEE Signal Process. Lett.*, vol. 28, pp. 1265–1269, 2021, doi: 10.1109/LSP.2021.3089437.

[20] Z. Wu, "ASVspoof 2015: The first automatic speaker verification spoofing and countermeasures challenge," in *Proc. Interspeech*, 2015, pp. 588–604, doi: 10.21437/Interspeech.2015-462.

[21] F. Xiao, Y. Lan, and J. Guan, *Segmenting Detection Strategy For Partially Fake Audio Detection*, 2022.

[22] J. Yi, "ADD 2022: The first audio deep synthesis detection challenge," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Singapore, May 2022, pp. 9216–9220.

[23] Z. Zhang, X. Yi, and X. Zhao, "Fake speech detection using residual network with transformer encoder," in *Proc. ACM Workshop Inf. Hiding Multimedia Secur.*, Jun. 2021, pp. 13–22.

[24] M. Todisco, X. Wang, V. Vestman, M. Sahidullah, H. Delgado, A. Nautsch, J. Yamagishi, N. Evans, T. Kinnunen, and K. Aik Lee, "ASVspoof 2019: Future horizons in spoofed and fake audio detection," 2019, *arXiv:1904.05441*.

[25] R. Reimao and V. Tzerpos, "FoR: A dataset for synthetic speech detection," in *Proc. Int. Conf. Speech Technol. Hum.-Comput. Dialogue (SpeD)*, Oct. 2019, pp. 1–10, doi: 10.1109/SPED.2019.8906599.

[26] J. M. Martín-Doías and A. Álvarez, "The vicomtech audio deepfake detection system based on Wav2Vec2 for the 2022 ADD challenge," in *Proc. IEEE Int. Conf. Acoustics, Speech Signal Process. (ICASSP)*, May 2022, pp. 9241–9245, doi: 10.1109/ICASSP43922.2022.9747768.

[27] H. Tak, M. Todisco, X. Wang, J.-w. Jung, J. Yamagishi, and N. Evans, "Automatic speaker verification spoofing and deepfake detection using wav2vec 2.0 and data augmentation," 2022, *arXiv:2202.12233*.

[28] Y. Xie, Z. Zhang, and Y. Yang, "Siamese network with Wav2Vec2 feature for spoofing speech detection," in *Proc. Interspeech*, 2021, pp. 4269–4273.

[29] X. Liu, F. Zhang, Z. Hou, L. Mian, Z. Wang, J. Zhang, and J. Tang, "Self-supervised learning: Generative or contrastive," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 1, pp. 857–876, Jan. 2023.

[30] Z. Cai, W. Wang, and M. Li, "Waveform boundary detection for partially spoofed audio," 2022, *arXiv:2211.00226*.

[31] X. Liu, M. Liu, L. Zhang, L. Zhang, C. Zeng, K. Li, N. Li, K. A. Lee, L. Wang, and J. Dang, "Deep spectro-temporal artifacts for detecting synthesized speech," in *Proc. 1st Int. Workshop Deepfake Detection Audio Multimedia*, Oct. 2022, pp. 69–75.

[32] C. Wang, J. Yi, J. Tao, H. Sun, X. Chen, Z. Tian, H. Ma, C. Fan, and R. Fu, "Fully automated end-to-end fake audio detection," in *Proc. 1st Int. Workshop Deepfake Detection Audio Multimedia*, Oct. 2022, pp. 27–33.

[33] H. M. J. Yi. (Jun. 9, 2022). *FAD: A Chinese Dataset for Fake Audio Detection | Zenodo*. Accessed: Aug. 18, 2022. [Online]. Available: https://zenodo.org/record/6635521#.Ysjq4nZBw2x

[34] M. Lataifeh and A. Elnagar, "Ar-DAD: Arabic diversified audio dataset," *Data Brief*, vol. 33, Dec. 2020, Art. no. 106503, doi: 10.1016/j.dib.2020.106503.

[35] *Every Ayah*. Accessed: Dec. 23, 2022. [Online]. Available: https://everyayah.com/

[36] N. Halabi, Ph.D. thesis, Univ. Southampton, School Electron. Comput. Sci., 2016, Accessed: May 21, 2023. [Online]. Available: http://en.arabicspeechcorpus.com/Nawar%20Halabi%20PhD%20Thesis%20Revised.pdf

[37] N. Halabi. *Arabic Speech Corpus*. Accessed: Dec. 11, 2022. [Online]. Available: http://en.arabicspeechcorpus.com/

[38] *Downloading Praat for Windows*. Accessed: Dec. 17, 2022. [Online]. Available: https://www.fon.hum.uva.nl/praat/download_win.html

[39] *GitHub—Nawarhalabi/Arabic-Phonetiser: Convert Arabic diacritised Text to a Sequence of Phonemes and Create a Pronunciation Dictionary From Them for Alignment Using HTK*. Accessed: Dec. 17, 2022. [Online]. Available: https://github.com/nawarhalabi/Arabic-Phonetiser

[40] M. Rizwan, B. O. Odelowo, and D. V. Anderson, "Word based dialect classification using extreme learning machines," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2016, pp. 2625–2629, doi: 10.1109/IJCNN.2016.7727528.

[41] M. Najafian, "Modeling accents for automatic speech recognition," in *Proc. 23rd Eur. Signal Proc. (EUSIPCO)*. Europ: University of Birmingham, 2013, p. 1.

[42] *King Saud University Arabic Speech Database—Linguistic Data Consortium*. Accessed: Dec. 24, 2022. [Online]. Available: https://catalog.ldc.upenn.edu/LDC2014S02

[43] GitHub. *GitHub—ARBML/Klaam: Arabic Speech Recognition, Classification and Text-to-Speech*. Accessed: Dec. 19, 2022. [Online]. Available: https://github.com/ARBML/klaam

[44] Y. Ren, C. Hu, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu, "FastSpeech 2: Fast and high-quality end-to-end text to speech," 2020, *arXiv:2006.04558*.

[45] K. Okabe, T. Koshinaka, and K. Shinoda, "Attentive statistics pooling for deep speaker embedding," 2018, *arXiv:1803.10963*.

[46] Z. Lv, S. Zhang, K. Tang, and P. Hu, "Fake audio detection based on unsupervised pretraining models," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2022, pp. 9231–9235.

[47] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*.

[48] W.-N. Hsu, B. Bolte, Y.-H. Hubert Tsai, K. Lakhotia, R. Salakhutdinov, and A. Mohamed, "HuBERT: Self-supervised speech representation learning by masked prediction of hidden units," 2021, *arXiv:2106.07447*.

[49] C. Lee and J. Glass, "A nonparametric Bayesian approach to acoustic model discovery," in *Proc. 50th Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, 2012, pp. 40–49.

[50] *Facebook/Hubert-Base-Ls960 · Hugging Face*. Accessed: Dec. 19, 2022. [Online]. Available: https://huggingface.co/facebook/hubert-base-ls960

[51] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, "Activation functions: Comparison of trends in practice and research for deep learning," 2018, *arXiv:1811.03378*.

[52] D. Ian, M. Michael, and C. Scott, *Bayesian Optimization Primer*. New York, NY, USA: Wiley, 2001.

[53] H. Wang, C. Ding, and H. Huang, "Multi-label linear discriminant analysis," in *Proc. 11th Eur. Conf. Comput. Vis.*, Heraklion, Greece: Springer, Sep. 2010, pp. 126–139.

[54] H. Hofbauer and A. Uhl, "Calculating a boundary for the significance from the equal-error rate," in *Proc. Int. Conf. Biometrics (ICB)*, Jun. 2016, pp. 1–4, doi: 10.1109/ICB.2016.7550053.

[55] R. Abdulhammed, H. Musafer, A. Alessa, M. Faezipour, and A. Abuzneid, "Features dimensionality reduction approaches for machine learning based network intrusion detection," *Electronics*, vol. 8, no. 3, p. 322, 2019, doi: 10.3390/electronics8030322.

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

[57] Z. Xie, I. Sato, and M. Sugiyama, "Stable weight decay regularization," Austria, Tech. Rep., 2020, p. 18.

[58] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. Cambridge, MA, USA: MIT Press, 2012.

[59] J. Gilmer, B. Ghorbani, A. Garg, S. Kudugunta, B. Neyshabur, D. Cardoze, G. Dahl, Z. Nado, and O. Firat, "A loss curvature perspective on training instability in deep learning," 2021, *arXiv:2110.04369*.

[60] B. Liu, W. Shen, P. Li, and X. Zhu, "Accelerate mini-batch machine learning training with dynamic batch size fitting," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2019, pp. 1–8, doi: 10.1109/IJCNN.2019.8851944.

[61] J. Brownlee, "What is the difference between a batch and an epoch in a neural network," *Mach. Learn. Mastery*, vol. 20, no. 1, 2018.

[62] A. Graves and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 1764–1772.

[63] T. Hori, S. Watanabe, and J. R. Hershey, "Joint CTC/attention decoding for end-to-end speech recognition," in *Proc. Proc. 55th Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, 2017, pp. 518–529.

ZAYNAB M. ALMUTAIRI received the B.S. degree in information technology from the College of Computer and Information Science, Majmaah University, Majmaah, Saudi Arabia, in 2017. She is currently pursuing the M.S. degree with the Department of Information Technology, College of Computer and Information Science, King Saud University, Riyadh, Saudi Arabia. Her research interests include deep learning, speech processing, speech recognition, data mining, and computer vision fields. She volunteered as a reviewer with IEEE Access and received four rewarding excellences regarding research output in the past few years.

HEBAH ELGIBREEN received the Ph.D. degree from King Saud University (KSU), in 2015. She specialized in artificial intelligence and machine learning with KSU. She is currently an Associate Professor with the Department of Information Technology, College of Computer and Information Sciences, KSU, where she is also the Director of the AI Center of Advance Studies. She is also leading the female branch of the Center of Smart Robotics Research, College of Computer and Information Sciences, KSU. Her research interest includes using ML approaches to improve collaborative robotics motions in shared environment. In the past couple of years, she was able to publish part of her work in different ISI journals. She is still ongoing with her project and looking for ways to integrate cognitive learning in order to apply innovative solutions that can be applied to more complex areas, including health and industry 4.0.