Expert Systems With Applications 171 (2021) 114591

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Review

Speaker identification through artificial intelligence techniques: A comprehensive review and research challenges

Rashid Jahangir [a,b,*], Ying Wah Teh [a,*], Henry Friday Nweke [c], Ghulam Mujtaba [d], Mohammed Ali Al-Garadi [e], Ihsan Ali [a]

[a] Department of Information Systems, Faculty of Computer Science and Information Technology, University of Malaya, 50603 Kuala Lumpur, Malaysia
[b] Department of Computer Science, COMSATS University Islamabad, Vehari Campus, Pakistan
[c] Department of Computer Science, Ebonyi State University, P.M.B 053, Abakaliki, Ebonyi State, Nigeria
[d] Center of Excellence for Robotics, Artificial Intelligence and Blockchain, Department of Computer Science, Sukkur IBA University, Sukkur 65200, Pakistan
[e] Department of Biomedical Informatics, Emory University, Atlanta, GA, USA

ARTICLE INFO

Keywords:
Speaker identification
Survey
Acoustic features
Artificial Intelligence
Deep learning
Speech databases

ABSTRACT

Speech is a powerful medium of communication that always convey rich and useful information, such as gender, accent, and other unique characteristics of a speaker. These unique characteristics enable researchers to recognize human voice using artificial intelligence techniques that are important in the areas of forensic voice verification, security and surveillance, electronic voice eavesdropping, mobile banking and mobile shopping. Recent advancements in deep learning and other hardware techniques have gained attention of researchers working in the field of automatic speaker identification (SI). However, to the best of our knowledge, there is no in-depth survey is available that critically appraises and summarizes the existing techniques with their strengths and weaknesses for SI. Hence, this study identified and discussed various areas of SI, presented a comprehensive survey of existing studies, and also presented the future research challenges that require significant research efforts in the field of SI systems.

1. Introduction

Speech signals are universal medium of communication that always carry rich and useful information such as accent, gender, emotion and other unique characteristics of a speaker. These unique characteristics which are also known as voice biometrics enable researchers to distinguish among speakers when calls are conducted over phones although the speakers might not be present physically. Through such characteristics, machines can become familiar with the utterances of speakers, similar to humans. Speaker Identification (SI) is a process of extracting the identity of a speaker by using machine according to the acoustic features of the given utterance.

Speaker Identification (SI) is an important domain of research for various applications, such as forensic voice verification for suspects detection (Campbell et al., 2009; Morrison et al., 2016), access control to various services like telephone network services (Hunt & Schalk, 1996), voice dialing, computer access control (Naik & Doddington, 1987), mobile shopping (Gomar, 2015) and mobile banking. Moreover, SI systems are widely used to improve security (Faundez-Zanuy, Hagmüller, & Kubin, 2007), automatic speaker labeling of recorded meetings (Arons, 1994), and personalized caller identification using intelligent answering machines (Schmandt & Arons, 1984).

The applications of SI can be divided into two categories as shown in Fig. 1. The first category depends on the existence of speaker's voice biometrics in the speakers' model which is further subdivided into open-set and closed-set. In an open-set SI the unknown speaker utterance is matched with the speaker model; the input speaker is rejected unless the exact match did not occur (Reynolds, 2002). On the other side, in closed-set, the unknown speaker utterance is matched with existing speakers' utterances in the model and the similarity score is calculated (Dutta, Patgiri, Sarma, & Sarma, 2015). Thus, a closed-set SI always return an output even though it may not be the accurate speaker. The second category of SI applications depends on the user control level, which is further categorized into text-independent and text-dependent SI. The

* Corresponding authors at: Department of Information Systems, Faculty of Computer Science and Information Technology, University of Malaya, 50603 Kuala Lumpur, Malaysia.
E-mail addresses: rashidjahangir@ciitvehari.edu.pk (R. Jahangir), tehyw@um.edu.my (Y.W. Teh), henry.nweke@ebsu.edu.ng (H.F. Nweke), mujtaba@iba-suk.edu.pk (G. Mujtaba), maalgar@emory.edu (M.A. Al-Garadi), ihsanalichd@siswa.um.edu.my (I. Ali).

https://doi.org/10.1016/j.eswa.2021.114591
Received 5 February 2020; Received in revised form 7 January 2021; Accepted 7 January 2021
Available online 12 January 2021

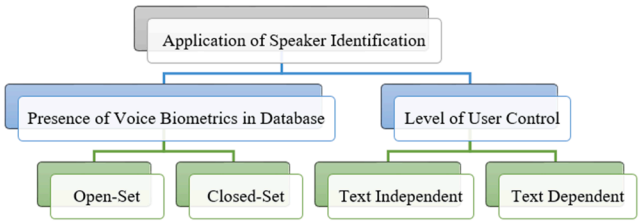

Fig. 1. Classification of speaker identification applications.

text-dependent SI requires the speakers precisely speaking the same utterance (Islam & Rahman, 2009; Kekre, Athawale, & Desai, 2011) whereas the text-independent SI system doesn't depend on the content of utterance. In other words, speakers can utter anything without any constraint (Revathi & Venkataramani, 2009; Verma, 2011). This type of system extracts discriminative features from speaker's pronunciation for construction of model and matches similarity between the features of an unknown speaker and those belonging to the model to determine "who is the speaker." The design and implementation of such systems is comparatively complex because of the extent of the sentences/words involved is broader. However, the versatility of the system is high, useful in real-time applications and room for its potential development is board.

Existing works in speaker identification defined the procedures for design and implementation of SI systems termed the automatic speaker identification pipeline. These include speech data collection, signal preprocessing and segmentation, acoustic feature extraction, dimensionality reduction, construction of classification model and evaluation of learning models. These procedures are presented in Fig. 2. As shown in Fig. 2, automatic speaker identification begins with speech data collection using various recording devices. These devices record voice differently because they own different sensors, sensitivities and recording capabilities. Another important phase is signal pre-processing that involves representation of raw speech signal. Generally, raw speech data includes a lot of spike and background-noise that lead to misclassification of learning model. Several methods such as noise reduction (Siam, El-Khobby, Abd Elnaby, Abdelkader, & Abd El-Samie, 2019), silence removal (Jahangir et al., 2020), pre-emphasis (An et al., 2019a), spectrogram representation (Wang, Xue, Wang, & Liu, 2020) and endpoint detection (Zhang, Shao, Wu, Geng, & Fan, 2020) have been proposed for speech data pre-processing in automatic speaker identification. These pre-processing methods are discussed in detail in Section 3.

Segmentation technique divides the speech signal into N number of frames using overlap or fixed window sizes to extract discriminative acoustic features. Window sizes play key role in mobile based implementation of speaker identification to minimize the computational time. The segmentation technique usually includes sliding window, events or energy based methods (Shi, Huang, & Hain, 2020). Feature extraction and feature selection derive relevant set of features to improve accuracy, reduce computation time and classification error. The extraction of acoustic features can be classified into shallow and deep features. Shallow features include the extraction of traditional handcrafted features such as time domain (statistical), frequency domain and ensemble empirical mode decomposition (EMD) features (Wu & Lin, 2009a, 2009b). Nevertheless, shallow features highly depend on domain expert knowledge, require large amount of labelled speech data, and employ dimensionality reduction techniques that are hard to generalizable. In recent years, automatic features extraction from raw speech data through DL (Tran & Tsai, 2020; Wang et al., 2020) were also proposed for automatic speaker identification to enhance classification performance. Deep learning techniques use high-level representation of data to extract discriminative features from raw speech data with various layers of neural nets and represent features from low-level to high-levels pyramid.

Deep learning techniques such as deep autoencoder, recurrent neural network and convolutional neural network are very popular techniques in pattern recognition, natural language processing (Sutskever, Vinyals, & Le, 2014), and now in automatic speaker identification. These techniques are discussed in details in Section 6. The features extracted from speech data coupled with machine learning algorithms are used to construct a speaker identification model. These machine learning algorithms include Gaussian Mixture Model (Al-Rawahy, Abdulnasir Hossen, & Ulrich Heute, 2012a, 2012b), Support Vector Machine (Faragallah, 2018), k Nearest Neighbor (Sardar & Shirbahadurkar, 2018b) and Artificial Neural Network (Wu & Tsai, 2011). Section 5 discusses the details of these classification algorithms. For DL, both features extraction and classification are trained for model construction. Finally, the automatic speaker identification system is evaluated using various performance metrics such as precision, recall and accuracy.

However, major research studies in automatic speaker identification concentrate on use of speech data uttered in a single language or accent, acoustic features and classifiers (Shahin, Nassif, & Hamsa, 2020) that are sometimes not effective to distinguish among speakers. To fully exploit speech data, acoustic features and classification algorithms for effective identification of speakers require fusion techniques. Fusion of speech data involves integration of recorded voices uttered in different languages through multiple recording devices to increase robustness,

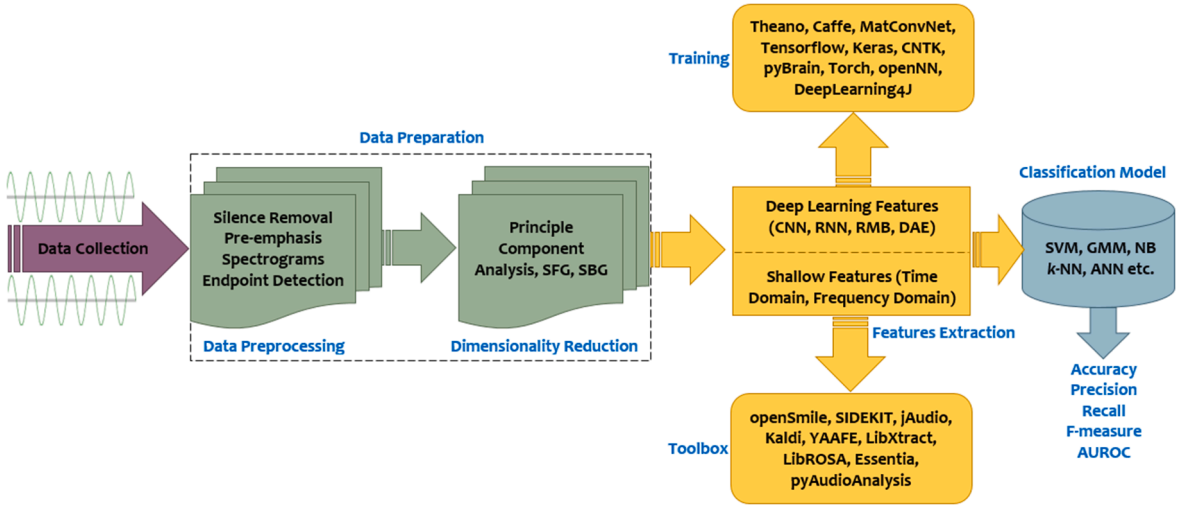

Fig. 2. Classical speaker identification process.

reliabilities and generalization ability of identification system.

Features extracted from speaker utterances are fused using different machine learning classifiers such as decision tree, support vector machine, and Gaussian Mixture Model to differentiate the speech data into high level of abstraction (Huang, Tian, Wu, & Zhang, 2019). Moreover, automatic feature extraction using deep learning to overcome the problems of temporal and spatial dependencies have now become progressive area of research in speaker identification (Yue & Yang, 2020; Zhang, Patras, & Haddadi, 2019). Furthermore, frequency and time domain features are fundamentally linear but only in real life, automatic speaker identification systems are nonlinear (Petry & Barone, 2002). Deep learning techniques automatically extract translational invariant features from speech data to reduce time spent on feature extraction methods.

Generally, fusion of different handcrafted features, multiple classifiers and deep learning systems have been proposed in automatic speaker identification in recent years (Calzà, Gagliardi, Favretti, & Tamburini, 2020; Fierrez, Morales, Vera-Rodriguez, & Camacho, 2018; Jahangir et al., 2020). For in-depth analysis, it is important to review these research articles. The aim of present study is to review automatic speaker identification systems that employed different machine learning classifiers and deep learning techniques for both feature extraction and classification. We present an inclusive review of recent advancements in automatic speaker identification. Particularly, we provide benchmark databases with their characteristics, numerous speech pre-processing techniques, feature fusion with traditional handcrafted features and feature extractor toolkits. Furthermore, we review various machine learning classifiers, deep learning approaches for constructing speaker identification model, evaluation measures to compute the performance of classification algorithms and extensively used deep learning implementation frameworks. The review taxonomy is shown in Fig. 3 and list of abbreviations used in this study with their full form is presented in Table 2. This study also discussed future research directions for improvements and attention based on the reviewed studies.

Several reviews and survey articles have been published in the area of speaker recognition in recent years (Larcher, Lee, Ma, & Li, 2014; Lawson et al., 2011; Saquib, Salam, Nair, Pandey, & Joshi, 2010). Conversely, these review articles focus on the traditional handcrafted features and machine learning classifiers for automatic speaker recognition. However, in this study, we not only focus on shallow features and machine learning classification but deep learning techniques for automatic feature extraction, benchmark databases, pre-processing techniques, feature selection methods for speaker identification. Recently, reviews on feature extraction methods were presented by (Disken, Tufekci, Saribulut, & Cevik, 2017; Tirumala, Shahamiri, Garhwal, & Wang, 2017). Tirumala et al. (2017) identified, compared, and analyzed various feature extraction methods and algorithms for speaker identification to provide a guide on feature extraction methods for speaker identification applications. However, our study extends beyond machine learning classifiers to include recent studies with deep learning techniques for speaker identification. Similarly, Disken et al. (2017) discuss feature extraction methods for speaker identification under noisy, channel mismatch and other degraded conditions. Finally, review that discuss deep learning algorithms and automatic feature extraction was presented by Tirumala and Shahamiri (2016) for automatic speaker identification. The study concentrated on the general deep neural network architecture and various phases of speaker identification. The review failed to present a detail explanation on substantial new proposed studies that focused on various deep learning algorithms such as RNN, CNN and DBN. A closely related review to this study is the one presented by (Tirumala et al., 2017) that discussed feature extraction methods and algorithms in speaker identification.

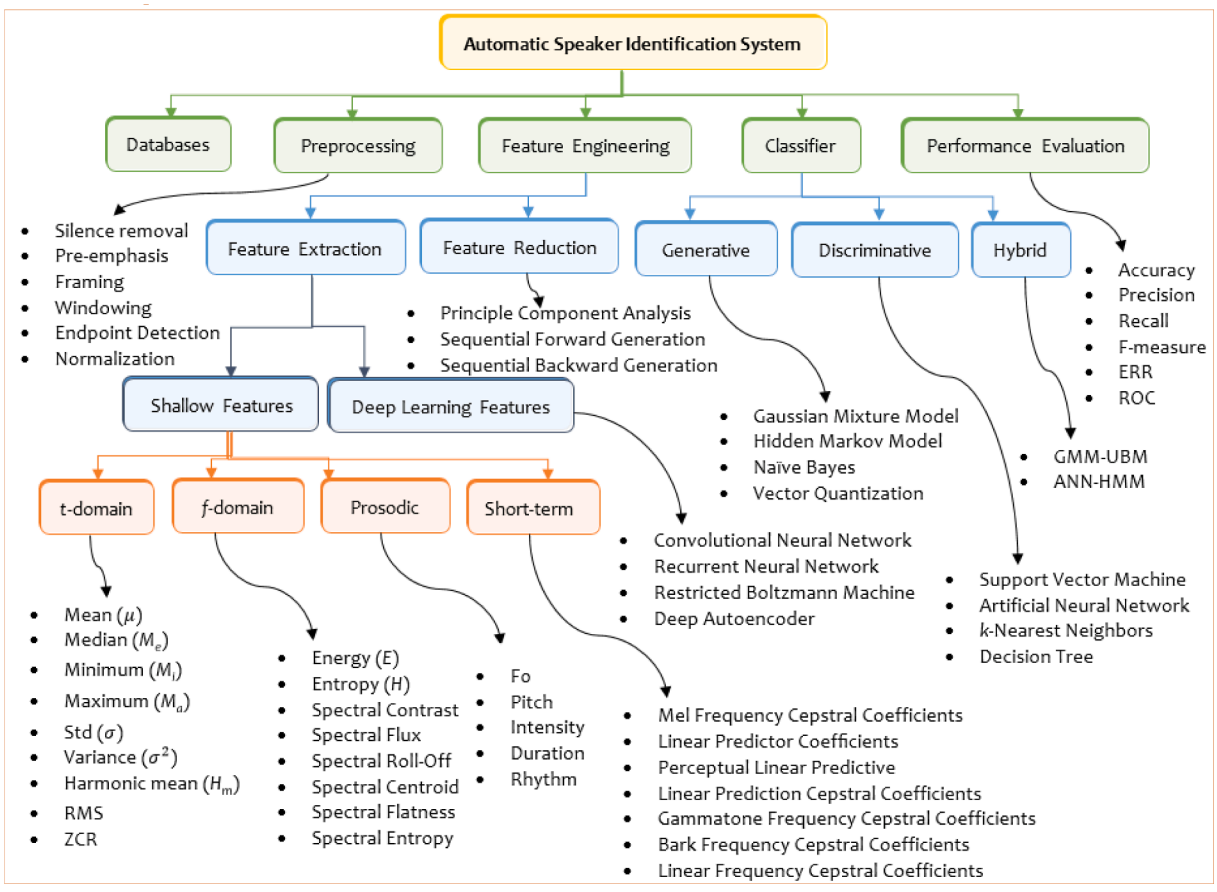

Fig. 3. Taxonomy of automatic speaker identification system.

The current study varies with their review in many aspects. Firstly, while their review presented only feature extraction techniques, the current study not only present the techniques but also provide a list of toolkits that can be used to extract these feature. Secondly, with the recent rise of deep learning techniques for automatic feature extraction, we provide deep learning techniques for feature extraction for automatic speaker identification. Finally, the current study also discusses various machine learning classifiers, benchmark databases and signal pre-processing techniques for automatic speaker identification. To the best of our knowledge, no comprehensive review is available in literature that explicitly discuss various components that are required to implement an automatic speaker identification system. To fill this gap, this study explores feature fusion and different classifiers in this significant area. Table 1 presents a list of recently published survey articles and compared according to the areas they covered.

The contributions of this paper are as follows:

- To summarize recent developments in databases construction, feature engineering, multiple machine learning classifiers system and deep learning for automatic speaker identification.
- To provide analysis on speaker identification techniques, strengths and weaknesses of these techniques.
- To highlight research gaps and future challenges in the developments of databases, feature engineering and classification models in speaker identification.

The remaining of this study is arranged as follows: Section 2 and Section 3 reviews databases for SI and speech signals processing respectively. Section 4 discusses various feature extraction and feature selection methods. Section 5 and 6 examines the ML and DL techniques with benefits and drawbacks. Section 7 describes the performance evaluation of ML and DL models for SI. Section 8 discusses DL implementation software frameworks. Section 9 reports the open issues, observations, and future research directions for improvements and attention. Finally, Section 10 concludes this review article.

2. Databases for speaker identification

This section describes the detailed analysis and review of different speech databases utilized for implementation and evaluation of SI systems. Though the provided inventory of SI databases (Table 3) may not be extensive, it includes the largest list in the current literature to best of our knowledge. The databases for SI have been built for multiple purposes and the variety of procedures makes it challenging for an impartial comparison of the speech corpora. In SI process, benchmark databases are critically important as low quality can affect the performance of proposed methods. Thus, according to Larcher et al. (2014) and Tirumala et al. (2017), the following relevant factors to be considered for benchmark databases:Table 4.

Table 2
. List of abbreviations with definitions.

Abbreviations	Definitions	Abbreviations	Definitions
SI	Speaker Identification	MFCC	Mel Frequency Cepstral Coefficients
DFT	Discrete Fourier Transform	IMFCC	Inverted Mel Frequency Cepstral Coefficients
FFT	Fast Fourier Transform	MFSC	Mel Frequency Spectral Coefficients
GMM	Gaussian Mixture Model	LFCC	Linear Frequency Cepstral Coefficients
k-NN	k-Nearest Neighbor	GFCC	Gammatone Frequency Cepstral Coefficients
SVM	Support Vector Machine	BFCC	Bark Frequency Cepstral Coefficients
ANN	Artificial Neural Network	LPCC	Linear Predictor Cepstral Coefficients
CNN	Convolutional Neural Network	LPC	Linear Predictor Coefficients
RNN	Recurrent Neural Network	PLP	Perceptual Linear Prediction
DNN	Deep Neural Network	PWCC	Perceptual Wavelet Cepstral Coefficients
RF	Random Forest	CFCC	Cochlear Cepstrum Coefficients
NB	Naïve Bayes	RASTA-PLP	Relative Spectra-Perceptual Linear Predictive
PCA	Principle Component Analysis	DWT	Dyadic Wavelet Transform
SFG	Sequential Forward Generation	WSBC	Wavelet Sub-Band Coding
SBG	Sequential Backward Generation	LBP	Local Binary Pattern
EM	Expectation Maximization	ZCR	Zero-Crossing Rate
FLDA	Fisher Linear Discriminant Analysis	BP	Back Propagation
UBM	Universal Background Model	CAE	Convolutional Variational Autoencoder
SGD	Stochastic Gradient Descent	ELU	Exponential Linear Rectification
RBM	Restricted Boltzmann Machine	IBP	Invariant Backpropagation
DBN	Deep Belief Network	ReLU	Rectified Linear Unit
LSTM	Long Short Term Memory	MOOC	Massive Open Online Courses
DAE	Deep Autoencoder	ResNet	Residual Neural Network
SAE	Stack Autoencoder	VGG	Visual Geometry Group

2.1. Language

The performance of SI may be influenced by the choice of language; as different languages illustrates different types of acoustic features. Although most of the SI databases presented in Table 3are built for English but the literature reports databases for several other languages

Table 1
. Recent reviews on SI and the topics they addressed, compared by the topics this study addresses. The comparison is made by including of speech preprocessing methods (Prep), databases (DB), features selection (FSel), features extraction toolkits (Libs), machine and deep learning methods (ML/DL), evaluation metrics (Metr) and DL implementation frameworks (Fworks).

Publication	Date	Prep.	DB	Feat.	FSel.	Libs.	ML	DL	Metr.	Fworks.
"Text-dependent speaker verification: Classifiers, databases and RSR2015" (Larcher et al., 2014)	2014	×	✓	✓	×	×	✓	×	×	×
"A review on Deep Learning approaches in Speaker Identification" (Tirumala & Shahamiri, 2016)	2016	×	×	×	×	×	✓	✓	×	×
"A Review on Feature Extraction for Speaker Recognition under Degraded Conditions." (Disken et al., 2017)	2017	×	✓	✓	×	×	✓	×	×	×
"Speaker identification features extraction methods: A systematic review" (Tirumala et al., 2017)	2017	×	×	✓	×	×	✓	×	×	×
This study	2020	✓	✓	✓	✓	✓	✓	✓	✓	✓

Table 3
. Characteristics of speech databases for speaker identification.

Corpus	Ethnic Groups	Lang	Size	Access	#Speakers female/male	Age Info	B/S	#CH	SR kHz	FRT	Source	Sesh
VoxCeleb	American British German Indian French	EN	153,516	Free[a]	1251 (563/688)	No	16	1	16	WAV	Quiet/ Noisy	–
RSR2015	Chinese, Malay, Unknown	EN	196,844	Free[b]	300 (143/157)	17–42	16	1	16	WAV	Quiet	9
TIMIT	American	EN	6300	Paid	630 (192/438)	No	16	1	16	WAV	Quiet	1
THCHS30	Chinese	ZH	13,389	Free[c]	40 (31/9)	19–55	16	1	16	WAV	Quiet	–
LibriSpeech train-clean-100	–	EN	11,373	Free[d]	251 (125/126)	No	16	1	16	FLAC	Quiet	–
ELSDSR	Danes, Icelander, Canadian	EN	198	Free[e]	22 (10/12)	24–63	16	1	16	WAV	Quiet	1
VCTK	British	EN	436,000	Free[f]	109 (62/47)	No	16	1	48	WAV		
CHAINS	IR, UK, USA	EN	–	Free[g]	36 (16/20)	No	16	1	44.1	WAV	Quiet	2
YOHO	USA	EN	18,768	Paid	138 (32/106)	No	16	1	8	WAV	Quiet	14
POLYCOST	BE, CH, DK, ES, FR, IR, IT, LI, NL, UK, PT, SE, TR	EN EU	1285	Paid	134 (60/74)	20–55	8		8	A-LAW	Quiet	5–14
Urdu Speech	Pakistani	UR	2500	Free[h]	10 (2/8)	No	16	1	16	WAV	Quiet	–
Switchboard	American	EN	2430	Paid	543 (241/302)	No	8	2	8	U-LAW	Quiet	1–25
CENSREC-4	Japanese	JA	8440	Paid	110 (55/55)	No	16	1	16	RAW	Quiet/ Noisy	–
JNAS	Japanese	JA	45,000	Paid	306 (153/153)	No	16	1	16	WAV	Quiet	–
GRID	British, Scottish, Jamaican	EN	340 00	Free[i]	34 (16/18)	18–49	16	1	25	WAV	Quiet	1–2
CMU	USA	EN	1158	Free[j]	84 (21/53)	No	16	2	16	RAW SPH MFC	Quiet/ Noisy	1–10
SIVA	Italian	IT	greater than2000	Paid	671 (336/335)	No	8	1	8	U-LAW	Quiet	3
LIEPA	Lithuanian	LT	greater than5000	–	376 (248/128)	No	16	1	22	WAV	Quiet/ Noisy	2
Turkish Speech	Turkish	TR	9000	Private	45 (20/25)	No	16	–	10	WAV	Quiet	1
KSU Arabic Speech	Saudis, Arabs & Non-Arabs	AR	–	Paid	269 (87/182)	No	16	2	48	WAV	Quiet/ Noisy	3
UT-Vocal Effort II	American	EN	–	–	112 (75/37)	No	16	1	44.1	–	neutral whisper	4
Fisher	American, Canadian	EN	16,454	Paid	100 (60/40)	Yes	8	2	8	U-LAW	Quiet/ Noisy	–
NOIZEUS	American	EN	720	Free[k]	6 (3/3)	No	16	1	25/8	WAV	Noisy	1

**Lang = Language, B/S = Bits per Samples, CH = Channel, SR = Sample Rate, FRT = Format, Sesh = Sessions, EN = English, ZH = Chinese, UR = Urdu, BE = Belgium, CH = Switzerland, DK = Denmark, FR = France, ES = Spain, IT = Italy, IR = Ireland, LI = Lituany, UK = United Kingdom, NL = Netherland, PT = Portugal, SE = Sweden, TR = Turkey.

[a] http://www.robots.ox.ac.uk/~vgg/data/voxceleb/.
[b] alarcher@i2r.a-star.edu.sg.
[c] http://www.openslr.org/18/.
[d] http://www.openslr.org/12.
[e] http://www2.imm.dtu.dk/~lfen/elsdsr/index.php?page = avl.
[f] https://datashare.is.ed.ac.uk/handle/10283/2651.
[g] https://chains.ucd.ie/ftpaccess.php.
[h] https://www.kaggle.com/hazrat/urdu-speech-dataset.
[i] http://spandh.dcs.shef.ac.uk/gridcorpus/.
[j] http://www.speech.cs.cmu.edu/databases/an4/.
[k] https://ecs.utdallas.edu/loizou/speech/noizeus/index.html.

as well. For instance, Chinese (Wang & Zhang, 2015), Urdu (Ali, Tran, Benetos, & Garcez, 2018), Japanese (Kawakami, Wang, Kai, & Nakagawa, 2014; Zhang et al., 2015), Italian (Falcone & Gallo, 1996), Hindi (Jawarkar, Holambe, & Basu, 2015) and Arabic (Alsulaiman, Muhammad, Bencherif, Mahmood, & Ali, 2013)

2.2. Demography

Number of speakers with its representatives such as gender and age are often seen as the key factors effecting SI systems. The size of speakers must be big enough as enhancement in accuracy of SI systems requires large-scale databases to ensure the classification rates are significant (Doddington, Przybocki, Martin, & Reynolds, 2000). The number of speakers listed in the given databases are still insufficient as only 9 of the 23 entries in the Table 3 have more than 200 speakers. Another disadvantage is the imbalanced distribution of gender as shown in Fig. 4. Out of the 23 databases for which information on gender is available, 11 can be called as gender-balanced with minimum 45% of speakers per gender while 4 databases comprise less than 30% of female speakers. Such imbalance is damaging as the accuracy of SI systems is considered to vary for both female and male speakers (Doddington, 2012). Moreover, age information is not always reported (Table 3) as it has been shown that identification between speakers is more difficult when the difference in age is small.

2.3. Lexical variability

The performance of SI systems is considered to be highly dependent on the quality of the input utterances. For example, a number of studies have been conducted to estimate the effect of utterance duration (Kanagasundaram, Vogt, Dean, Sridharan, & Mason, 2011; Vogt, Lustri, & Sridharan, 2008). Other studies have reported that the identification depends on utterance contents that were utilized for both training and testing, which means that, for fixed utterance duration, various parts may not be equally beneficial for SI (Kahn, Audibert, Bonastre, & Rossato, 2011; Nosratighods, Ambikairajah, Epps, & Carey, 2010). In case of

Table 4
. Pre-processing methods used in reviewed studies.

Methods	Strengths	Weaknesses	Reference
Silence Removal	Reduce processing time and enhance the performance of SI system	Only suitable for white Gaussian noise	Avci (2009), Daqrouq (2011), Daqrouq and Tutunji (2015), Fan and Hansen (2010), Jawarkar et al. (2015), Lukic et al. (2016), Medikonda and Madasu (2018), Novotný et al. (2019), Sarma and Sarma (2013), Wu and Tsai (2011), Zhao et al. (2014)
Pre-emphasis	Negative spectral slope of the various voiced parts is improved which enhance the performance.	gain of low frequencies also become high.	An et al. (2019a), Avci (2009), Faragallah (2018), Liu, Wu, Li, Li, & Shen (2018), Medikonda and Madasu (2018), Renisha and Jayasree (2019), Sun, Gu, Xie, and Chen (2019)
Hamming Window	reduces errors in the estimation of distortion	huge variance of spectral estimation	Al-Rawahy et al. (2012a, 2012b), Avci (2009), Faragallah (2018), Liu et al. (2018), Mporas et al. (2016)
End-Point Detection	incorporate spectral information and minimize the computational resources	In low SNR and non-stationary environments, end-point detection often fails and performance of SI degrade dramatically	Avci (2009), Renisha and Jayasree (2019)
Speech Signals Normalization	Prevent an error estimate caused by a change in the volume of speakers.	increase in the magnitude of jitter (precision measure for displacement estimation)	Avci (2009), Daqrouq (2011), Daqrouq and Tutunji (2015), Renisha and Jayasree (2019), Wu and Lin (2009a)
Median Filtering	Reduces random noise, particularly when noise amplitude likelihood density has large tails.	For multidimensional signals it does not produce satisfactory results as compared to one-dimensional	Sarma and Sarma (2013)
Spectrogram	Express speech signal by combining the benefit of both time and frequency domains and represents the relationship of frequency, time, and energy amplitude directly.	poor frequency and time resolution	An et al. (2019a), Bunrit, Inkian, Kerdprasop, and Kerdprasop (2019), Imran, Haflan, Shahrebabaki, Olfati, and Svendsen (2019), Liu et al. (2018), Lukic et al. (2016), Yadav and Rai (2018)
Recursive Least Squares	small memory requirement, converge faster and easily changed into real-time systems	computationally intensive, prevent the classifier from going to sleep;	Dhakal et al. (2019)

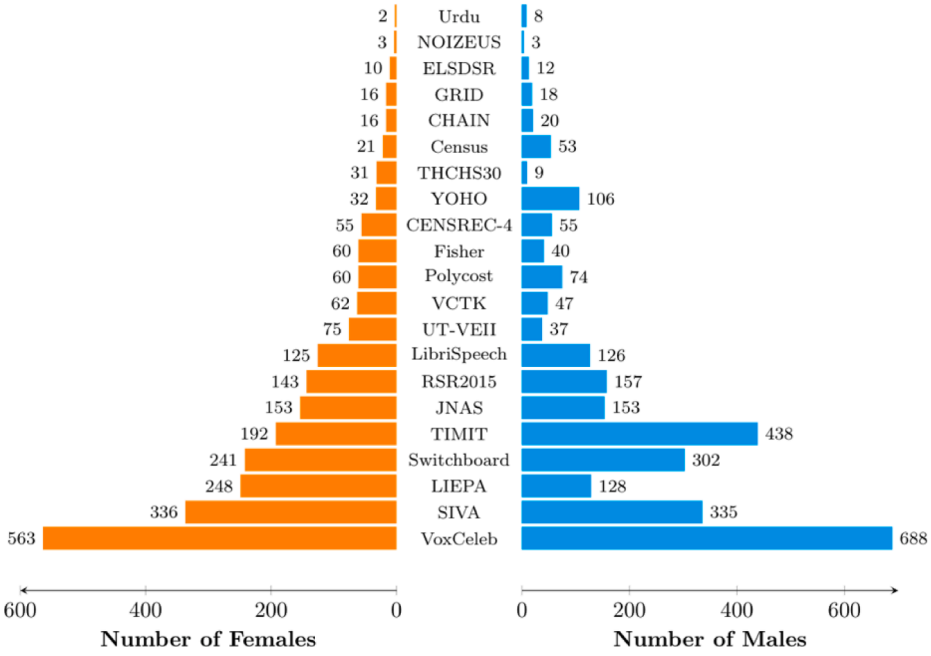

Fig. 4. Number of subjects per gender comparison.

text-dependent SI where both training and testing utterances are unchanged, lexical content is very important because it can influence the performance of the system. Thus, effect of the lexical-content must be examined when implementing a text-dependent SI system.

2.4. Session variability

Session variability factors such as recording time or device, channel mismatch, environment and ambient noise can influence the performance of text-dependent SI systems. Because of the cost and the complexity of data acquisition, most databases were collected under controlled environment and using the single microphone, which significantly reduces the channel and noise variation across sessions (e.g. ELSDSR, YOHO). Other databases (Nagrani, Chung, & Zisserman, 2017) concentrate on adverse conditions by offering speakers recordings in different environments like quiet studio interviews, outdoor stadiums, red carpet (e.g. VoxCeleb). Lastly, some databases (Hennebert, Melin, Petrovska, & Genoud, 2000a; Petrovska, Hennebert, Melin, & Genoud, 1998) provide explicit channel disparity by speakers recorded on various devices but don't enforce any background noise or environmental factors during recording (e.g. Polycost). The count of sessions is often limited because of the recording costs, which are proportional to the duration and the number of times a speaker needs to be mobilized. Among the 23 databases listed in the Table 3, only 5 (Campbell &

Higgins, 1994; Godfrey, Holliman, & McDaniel, 1992; Hennebert, Melin, Petrovska, & Genoud, 2000b; Kominek & Black, 2004; Larcher et al., 2014) have more than 5 sessions.

2.5. Speech recording devices

Different devices record voice differently because they own different sensors, sensitivities and recording capabilities. For instance, some of the microphones are built to record speech for a specific environment. In addition, microphones are also designed for a specific purpose such as noise cancelation microphones, computer microphones, microphones fixed in smartphones and portable microphones.

The details of different benchmark speech databases utilized by various research studies for speaker identification is presented below.

VoxCeleb (Nagrani et al., 2017) is a large-scale SI database derived from YouTube videos. VoxCeleb is an English language-based speech database that includes 153,516 utterances of variable length belonging to 563 females and 688 male celebrates. The speakers cover a wide variety of accents, ethnicities, ages, professions. Speech utterances comprised in the database are recorded in a various challenging acoustic environment such as outdoor stadium, red carpet, quiet studio interviews, speeches made for large audience and extracts from professionally recorded multimedia. Crucially, all of them are degraded with laughter, background chatter and real world noise, room acoustics, overlapping speech, channel noise and quality of recording devices.

TIMIT (Garofolo et al., 1993) speech corpus has been built for the development of automatic speech recognition (SR) systems. It includes 6300 utterances, ten utterances uttered by each speaker from eight major dialect regions (New York City, New England, Northern, South Midland, North Midland, Southern, Army Brat, Western) of the USA. The dialect region of the speaker is the geographical zone of the United States where they lived in their childhood. Moreover, the database is gender-unbalanced with 70% male speakers.

THCHS30: (Wang & Zhang, 2015) is a Chinese speech corpus, build for speech and speaker recognition systems. This corpus comprises around 30 h of voice recorded through one carbon microphone under silent condition. The utterances are divided into four groups: A (sentence ID 1–250), B (251–500), C (501–750), D (751–1000). The utterances of the first three groups are integrated as a training set, which contains 30 participants and 10,893 utterances. The utterances in last group are used as a test set, which contains 10 participants and 2,496 utterances. All the utterances are is recorded at a sample rate of 16 kHz and bit rate of 16 bits.

LibriSpeech: (Panayotov, Chen, Povey, & Khudanpur, 2015) corpus is publicly available and is prepared from audiobooks of the LibriVox to develop automated speech recognition and speaker identification models employing state-of-the-art ML and DL techniques. LibriSpeech includes English speeches related audio files that belong to male and female English speakers. All the utterances in this dataset are sampled at 16 kHz frequency and bit rate of 16 bits. This corpus includes five different training and testing sets for developing an SI model.

ELSDSR (Feng & Hansen, 2005) is an English language voice database, designed for the implementation and evaluation of SI and accent recognition systems. It was recorded in single session from 10 female and 12 male students and researchers at chamber building of Technical University of Denmark (DTU). The utterances were recorded using MARANTZ PMD670 recorder into the most regularly used file format-.wav (PCM) at sampling rate of 16 kHz and a bit rate of 16 bits. In addition, the corpus is divided into 154 (7*22) training utterances and 44 (2*22) test utterances.

Urdu Speech Corpus (Ali et al., 2018) is an Urdu language speech database, designed for both speaker and speech recognition tasks. It contains 2500 audio files of different length belonging to 8 male and 2 female individuals. As the quality of speech depends on the bit rate, sample rate (sampling frequency), file format and encoded method (MicroPyramid, 2011), all the speech utterances are consisting of mono-channel, bit size of 16 per sample, sample rate of 16 kHz(preferred) and encoded in wav format. In addition, this database is publicly available for research in this area.

RSR2015 (Larcher et al., 2014) was designed to provide the research community with a massive database of gender-balanced speakers. RSR2015 contains 196,844 audio files recorded in nine sessions with several tablets and hand-phones from 143 female and 157 male speakers. A close attention was paid to the lexical-content so as to make impartial comparison of text-dependent SI systems under various lexical constraints.

CHAINS: (Cummins, Grimaldi, Leonard, & Simko, 2006) is a conventional speech corpus recorded under different speaking styles including normal, whisper, fast etc. from a same speaker for every recorder sentence. Additionally, the corpus was recorded within one dialect for forensic SI. However, it also includes few speakers from out-of-dialect for comparison and features 36 speakers in total. Out of these 36 speakers, 28 (14 females, 14 male) were from the Ireland and 8 (4 females, 4 male) were from the USA or UK. Each recorded sentence in corpus is around 1– 2 s which is freely accessible for research purposes.

POLYCOST (Hennebert et al., 2000a; Petrovska et al., 1998) is a telephony speech database recorded in different European languages. The 60 female and 74 male speakers from different countries of Europe uttered connected digits in English and sentences in English and mother tongue. The database covers 6 sessions per speaker and was recorded in office and room environments.

Technical information presented in Table 3 was extracted from existing studies. However, for some databases matlab function audioinfo was used to get the utterance technical content. From this table, it can be concluded that most of the databases use sampling frequency of 16 kHz, mono channel, WAV format and 16 bits per sample.

The investigation of reviewed studies revealed that TIMIT was the most commonly used database for SI. As shown in Fig. 5 nine studies (Ajmera, Jadhav, & Holambe, 2011; Al-Rawahy et al., 2012a, Al-Rawahy et al., 2012b; Krobba, Debyeche, & Selouani, 2019; Lukic, Vogt, Dürr, & Stadelmann, 2016; Mporas, Safavi, Gan, & Sotudeh, 2016; Renisha & Jayasree, 2019; Sadıç & Bilginer Gülmezoğlu, 2011; Zhang, Zou, Sun, & Wu, 2018) employed TIMIT to train and evaluate the performance of proposed model. TIMIT is popular due to the phonetically aligned and transcribed, lexically, syntactically and phonetically representative, compact and easily available. After the TIMIT, the most employed databases were VoxCeleb (Hajavi & Etemad, 2019; Jung, Heo, Yang, Shim, & Yu, 2018; Yadav & Rai, 2018), NIST-SRE (Almaadeed, Aggoun, & Amira, 2016; Ghahabi & Hernando, 2018; Zhao, Wang, & Wang, 2014), ELSDSR (Abdalmalak & Gallardo-Antolín, 2018; Al-Rawahy et al., 2012a, Al-Rawahy et al., 2012b; Dhakal, Damacharla, Javaid, & Devabhaktuni, 2019; Soleymanpour & Marvi, 2017), CHAIN (Manikandan & Chandra, 2016; Sardar & Shirbahadurkar, 2019; Wang et al., 2015), YOHO (Almaadeed et al., 2016; Chakroborty & Saha, 2009; Michalevsky, Talmon, & Cohen, 2011) and Switchboard (Medikonda & Madasu, 2018; Zhang, Koishida, & Hansen, 2018b). Moreover, the Fig. 5 shows that several studies have employed benchmark databases and 18 have recorded their own exclusive databases because of the language, environment, size constraint of publicly available benchmark database. Furthermore, they produced their own databases for any specific type of application. For instance, (Ali et al., 2018) recorded his own database in Urdu language. However, the SI systems must be evaluated on benchmark databases to show the effectiveness of the proposed models.

3. Speech pre-processing and segmentation

In this section, we present a survey of different pre-processing techniques that were utilized by researchers in various research areas of speaker identification. Speech signals pre-processing is very critical phase in the systems where background-noise or silence is completely undesirable. Systems like SI and Speech Recognition (SR) requires efficient feature extraction approaches from speech signals where most of

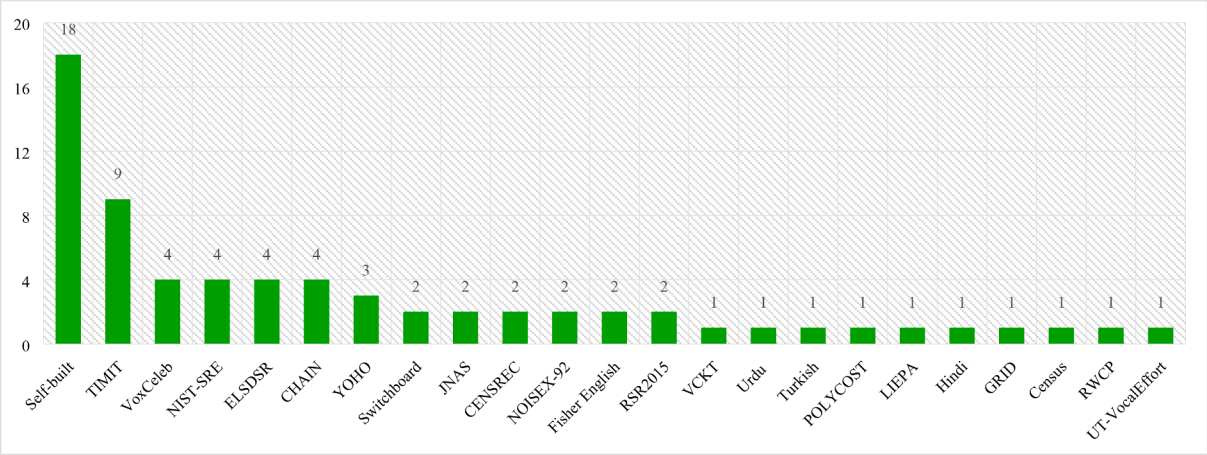

Fig. 5. . The frequency of speech databases in the reviewed studies.

the spoken portion includes speaker-related attributes. The details of different pre-processing techniques utilized by various research studies of speaker identification are presented below.

3.1. Silence removal

The speech signal can contain silence at various positions such as the beginning of signal, between the words of the sentence and at the end of signal. The removal of unspoken parts from speech signals reduces computational time and complexity. It is therefore necessary to remove unspoken parts before further processing. Unspoken parts of speech signal are eliminated by first labelling each sample as spoken/unspoken using statistical properties of background-noise (Jasmine, Sandhya, Ravichandran, & Balasubramaniam, 2016; Saha, Chakroborty, & Senapati, 2005). The mean and standard deviation of each sample of the given utterance is calculated as:

$$\mu = \left(\frac{1}{N}\right)\sum_{k=1}^{N} x(t) \tag{3.1}$$

$$\sigma = \sqrt{\left(\frac{1}{N}\right)\sum_{k=1}^{N} (x(t)-\mu)^2} \tag{3.2}$$

where $x(t)$ represent voice signal, μ is the mean and σ is the standard deviation. Background noise is characterized by Eqs. 3.1 and 3.2. For each sample, if one-dimensional Mahalanobis distance function i.e. $(|x-\mu|/\sigma) \geq 3$ then the sample is treated as spoken sample, otherwise it is a silence/unvoiced sample.

In their recent study, Novotný, Plchot, Glembek, and Burget (2019) used voice activity detection based on BUT Speech phoneme recognizer[1], as presented in (Matejka, Burget, Schwarz, & Cernocky, 2006), and removed each frame that was labeled as silence. Similarly, Medikonda and Madasu (2018) disregard the silent frames via voice activity detection (Sohn, Kim, & Sung, 1999) and extracted speaker specific features from voice frames.

3.2. Pre-emphasis

Pre-emphasis refers to the filtering that highlights the high frequencies of speech signal. It is used to align the spectrum of uttered voices that have a sharp roll-off in the higher frequency region. For uttered voices, the glottal source has a slope of roughly -12dB/octave (Rabiner, 1989). In comparison, when acoustic energy emanates from the lips, this triggers an approximately $+6$ dB/octave rise to the spectrum. Consequently, when a voice is from a distance by microphone, has a downward slope of roughly -6 dB/octave related to the true vocal tract spectrum. Therefore, pre-emphasis eliminates some glottal effects from the uttered voice. The most widely employed pre-emphasis filter is equivalent to

$$H(z) = 1 - \alpha z^{-1} \tag{3.3}$$

where α controls the pre-emphasis filter slope and its value varies between $[1, -0.97]$.

Pre-emphasis filter inevitably changes distribution of energy across frequencies along with overall energy level. This could have critical impact on the acoustic features related to energy (Zhao & Wang, 2013).

3.3. Framing

Signal framing is a process of dividing continuous speech signal into segments of fixed length as shown in Fig. 6. As the signal is non-stationary, speaker characteristics can change during speech. However, speech signals are assumed stationary for shorter period of time (20–30 ms). By framing the signal, this stationary can be estimated and local acoustic features can be extracted. In addition, the information between two successive frames can be preserved by overlapping these frames. Usually, a frame is generated with 25 ms duration and 10 ms overlap between consecutive frames.

3.4. Windowing

As speech signal is non-stationary in nature, where statistical properties (changes in prosody, spectral features and random variability of vocal tract) are not constant over time, therefore, it is impossible to make the use of DFT. For most phonemes, the statistical properties of speech signal remain constant (i.e. stationary) for a short time span (10–20 ms). Traditional signal processing techniques can therefore be implemented effectively for short window of time which is called frame of N samples (Gill, Kaur, & Kaur, 2010; Togneri & Pullella, 2011). In most of the speech processing systems, signals are divided into overlapping frames before extracting features. In order to reduce spectral distortions, all frames are multiplied by a Hamming window using Eq. 3.4.

$$w(n) = 0.54 - 0.46\cos\left(\frac{2\pi n}{N-1}\right), 0 \leq n \leq N \tag{3.4}$$

N is the number of samples in each frame. The output of the windowed

[1] https://speech.fit.vutbr.cz/software/phoneme-recognizer-based-long-temporal-context

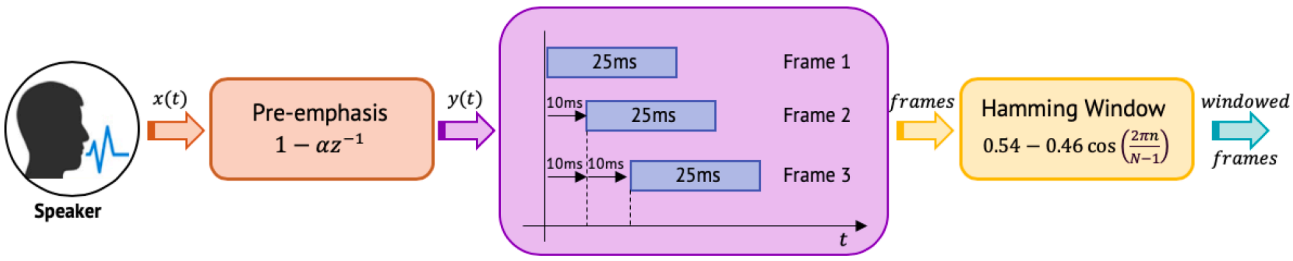

Fig. 6. Speech Signal Framing and Windowing (Jahangir et al., 2020).

speech signal becomes:

$$y(n) = x(n)w(n) \tag{3.5}$$

The other most commonly used windowing function are Hanning, half parallelogram and triangle.

3.5. Endpoint detection

It is a process of separating the segments of speech signal from background-noise (Sahoo & Patra, 2014). The background-noise is known to be white-noise and therefore its distribution is normal distribution (Blazek & Hong, 2012; Bou-Ghazale & Assaleh, 2002; Saha et al., 2005). Mathematically,

$$p(x) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\frac{1}{2} \left(\frac{x-\mu}{\sigma} \right)^2 \right) \tag{3.6}$$

The parameters σ and μ in above equation are computed using initial 3200 speech samples of an utterance as these samples are known to be background-noise. So, for any speech sample$x$ if $(|x-\mu|/\sigma) \leq 3$ then it means that x belongs to the white-noise distribution and therefore it can be removed from the utterance.

3.6. Speech signal normalization

Normalization (Daqrouq, 2011) makes the speech signals comparable irrespective of variations in magnitude by using Eq. 3.7.

$$S_{Ni} = \frac{S_i - S}{\sigma} \tag{3.7}$$

where S_i is the ith part of signal S, σ and S are the standard deviation and mean of S respectively, S_{Ni} is the normalized ith part of signal S.

3.7. Spectrogram

Speech signals are transformed into spectrograms by using short term Fourier transform in order to extract discriminative features automatically (Badshah et al., 2019; Stolar, Lech, Bolia, & Skinner, 2017; Sun, Chen, Xie, & Gu, 2018). A spectrogram is a two-dimensional visual representation where x-axis represents the time, while the frequency is shown on the y-axis, and the color of each point represents the energy amplitude at specific time. To generate a spectrogram, a speech signal is divided into frames which are multiplied by a Hamming window as shown in Fig. 6. Each windowed frame is transformed into the frequency-domain from time-domain by applying fast Fourier transform (FFT). Afterwards, band-pass filters are applied to the frequency-domain segment and Mel-scale is used to distribute the center frequencies of each filter. Next, a logarithmic function is performed on the output of each band-pass filter to suppress the dynamic range. Finally, the output of each discussed step is assembled frame by frame to generate the spectrogram of the speech signal.

4. Feature engineering methods for speaker identification

Feature engineering is one of the key steps in all classification problems. The process of feature engineering include three phases namely; feature extraction, representation of features and feature selection (Mujtaba et al., 2019). The resulting feature vector (in numeric form) of the feature engineering step is fed as an input into machine learning algorithms (*k*-NN, SVM, NB, etc.) for the construction and validation of classification model. The detailed explanation and review of feature extraction and feature selection methods in speaker identification are given in the following subsection.

4.1. Feature extraction techniques

Voice signals contain information about the human auditory system and human voice. It is critical for the extracted acoustic features to offer sufficient data that fit with SI modelling appropriately. Therefore, several researchers have exerted considerable effort to improve the effectiveness of the SI system through the extraction of robust and relevant features. Several methods and algorithms are stated in the literature for extracting the features from speech signals. The voice features can be classified into two categories namely handcrafted features and automatic features through deep learning (Georgescu, Ionescu, & Popescu, 2019; Wang, 2020). The handcrafted features can be further classified into time-domain features (*t*-domain features) and frequency-domain features (*f*-domain features) (Nweke, Teh, Mujtaba, Alo, & Al-garadi, 2019). The detailed review of these features is given below.

4.1.1. Handcrafted features

In automatic speaker identification and analysis of time series data, feature extraction process plays an important role to minimize the computational time and complexity particularly for mobile based implementation (Tiwari, Hashmi, Keskar, & Shivaprakash, 2020; Yue & Yang, 2020). Feature extraction and selection drive set of feature vectors that reduce identification errors, and to identify the most discriminative features for speaker identification tasks. Speaker identification based handcrafted features can be categorized into three types. These include *t*-domain features, *f*-domain features, prosodic features, acoustic features and cepstral domain features.

4.1.1.1. Time-domain features. *t*-domain features extract mathematical or statistical measures from raw speech signals in order to represent speaker utterance characteristics. The time-domain features extracted from raw speech signals are a degree of variation, measure of central tendencies and distribution of the signal shape.

I. *Measure of variability:* These features describe the degree to which the speech signals are dispersed between the central points over a distance. The higher degree of variation shows the worse distribution of raw speech signals. The feature derived as degree of variation are standard deviation (σ), variance (σ^2), interquartile range (I_r), coefficient of variation (C_v), root mean square (R_{ms}),

signal magnitude area, pitch angle (P_k), peak amplitude (P_a) and signal power (S_p). Measure of variation based time-domain features requires less computational cost and significant to determine the probability distribution of speech signals (Figo, Diniz, Ferreira, & Cardoso, 2010; Ouyang, Sun, Chen, Yue, & Zhang, 2018).

II. *Measure of central tendencies:* These set of features represent how the speech signal is related to the central points and describe the location of central values. Features derived as measure of central tendency contain minimum (M_i), maximum (M_a), mean (μ), median (M_e) and harmonic mean (H_m) of every 3-axis of the speech signal. These features have low computational time to process with minimum computation requirement. Moreover, these features have demonstrated significant improvement in performance for pattern recognition and estimation of energy expenditure (Nagori, 2016).

III. *Distribution of shape:* Features based on signal helps to understand the distribution and shape of the raw speech signal. Distribution of shape based features are kurtosis and skewness. Kurtosis determines the spike or flat of the speech signal distribution while skewness calculates the asymmetric probability distribution of speech signals. These features have demonstrated impressive results in speech recognition, speaker identification and related applications (Nakamura, 2002; Nweke, Teh, Mujtaba, & Al-Garadi, 2019).

All the time-domain features are listed in Table 5 with their formula used to measure each feature.

4.1.1.2. Frequency-domain features. Frequency-domain features (or spectral features) represent the dispersal of signal energy and are mainly used to identify the repetitive nature of speech signals. The raw speech signals (time-based signal) are converted into the frequency-domain by Fast Fourier Transform (FFT) function. From the transformed *f*-domain data, various features can be extracted such as spectral entropy, spectral centroid, spectral flux and mean frequency. According to Shannon (2001) spectral entropy calculates the quantity of information contained in a speech signal. In order to measure the compressibility, Shannon (2001) proposed Boltzmann Eq. (4.1); more information means less compression.

$$1 \Big/ \log N \left(\sum_{i=1}^{N} P(Z[i]) \log P(Z[i]) \right) \tag{4.1}$$

where N is the number of values for signal Z, and Z_i is the ith value.

All the frequency-domain features are listed in Table 6 with their formula used to measure each feature.

4.1.1.3. Prosodic features. Prosodic features (or long-term features) are non-segmental part of speech confined in long utterances, for instance prosodic intonation is a term commonly used to describe the variations in energy (intensity), pitch, rhythm and stress. These features significant for stress-times languages such as English and Arabic (Shahin, Epps, & Ahmed, 2016). Moreover, prosodic features investigate the acoustic qualities of speech signal and give information about the speaker. The accent in these features plays an important role in the identification of speaker. The different prosodic features Table 7 that can be derived from the speech signal are fundamental frequency, pitch, intensity (energy), formants, duration and rhythm (Ajmera et al., 2011; Sekkate, Khalil, & Adib, 2019). Here, the first feature pitch is also known as fundamental frequency (F0). F0 is the proportion of vocal fold vibrations for the duration of voice phonation (Tirumala et al., 2017) while Pitch is integral to the periodic human speech signal and represents the perceived pitch. It is important to note that F0 and pitch are considered same when refereeing to frequency of vocal fold vibration in the literature, however they are different characteristic of speech signal (Kinnunen, 2003). The existing studies highlights that these features fused with short-term spectral features are found to be the discriminative features for automatic speaker identification (Jagdale, Shinde, & Chitode, 2020).

4.1.1.4. Short term spectral features. Short-term spectral features are extracted from short frames (20–30 ms) of speech signals, since the speech signal is continuously changing as an articulation effect. As a result of these short frames, the extracted features are perceived to be stationary and preserve the local information. Mel frequency cepstral coefficients (MFCCs), linear prediction coefficients (LPC) and Perceptual Linear Prediction (PLP) are the most widely employed short-term spectral features in speaker identification.

MFCCs feature is a set of coefficients that are calculated by using frequencies of vocal track information. In cepstral domain, MFCC features represent speech signals that employ Discrete Fourier Transform (DFT) to reflect windowed short-term signals as real cepstrum of speech signal. MFCCs frequency bands are equally distributed on Mel scale as these features are driven from our natural system of auditory perception (Shahamiri & Salim, 2014).

The MFCCs feature extraction method mainly comprises frame blocking (dividing the speech signal into short frames), windowing the speech signal, calculation of magnitude spectrum applying the Fourier Transform, logarithm of magnitude spectrum, warping the frequencies on Mel-scale and applying the inverse discrete cosine transform (DCT) as shown in Fig. 7. The detailed explanation of these steps is presented below.

- *Frame blocking and windowing*: As discussed in Section 3.3, the speech signal is non-stationary in nature or slowly time-varying, speaker features can change during speech. For stable acoustic features, speech signal needs to be investigated over a shorter period of time. Therefore, a

Table 5
List of time-domain features.

Feature	Formula	Feature	Formula
Mean (μ)	$\bar{s} = 1/N\sum_{i=1}^{N} s_i$	Root mean square (R_{ms})	$rms = \sqrt{1/N\sum_{i=1}^{N}(s_i)^2}$
Median (M_e)	$median_i(s_i)$	Peak amplitude (P_a)	$\max(s_i) - \min(s_i)$
Minimum (M_i)	$min_i(s_i)$	Pitch angle (P_k)	$arctan\left(x_i/\sqrt{y^2 - x_i^2}\right)$
Maximum (M_a)	$max_i(s_i)$	Kurtosis (K_r)	$E\left[\left(s_i - \bar{s}\right)^4\right] \Big/ E\left[\left(s_i - \bar{s}\right)^2\right]^2$
Standard deviation (σ)	$\sigma = \sqrt{1/N\sum_{i=1}^{N}(s_i - \bar{s})^2}$	Skewness (S_k)	$E\left[\left(s_i - \bar{s}/\sigma\right)^3\right]$
Variance (σ^2)	$\sigma^2 = \frac{\sum (s_i - \bar{s})^2}{N}$	Signal power (S_p)	$\sum_{i=1}^{N} s_i^2$
Harmonic mean (H_m)	$1/N\sum_{i=1}^{N} 1/s_i$	Coefficient of variation (C_v)	σ_{si}/μ_{si}
Interquartile range (I_r)	$Q_3(s_i) - Q_1(s_i)$	Zero Cross Rate (ZCR)	$1/N\sum_{i=1}^{N} \lvert s_i - s_{i-1} \rvert$

Table 6
List of frequency-domain features.

Feature	Description	Formula	References
Energy (E)	The energy of the signal reflects the area under the square magnitude.	$\sum_{i=1}^{N}[s_i]^2/\boldsymbol{length}(s_i)$	(Yakovenko & Malychina, 2016)
Mean frequency (μF)	Mean frequency measures the mean normalized frequency of the power spectrum of a speech signal.	$\sum_{i=1}^{N}(\boldsymbol{is}_i(\boldsymbol{F}))/\sum_{j=1}^{N}\boldsymbol{s}_j(\boldsymbol{F})$	(Almaadeed et al., 2016)
Spectral Flux	Spectral Flux differentiate between normalized spectral magnitudes.	$\sum_{i=1}^{N}\left(Z^{(k)}[i]-Z^{(k-1)}[i]\right)^2\boldsymbol{Z}^{(k)}$ and $\boldsymbol{Z}^{(k-1)}$ are the normalized magnitudes of Fourier transform at $\boldsymbol{k}$ and $\boldsymbol{k}-1$ frames.	(Mannepalli, Sastry, & Suman, 2017)
Spectral Roll-Off	Spectral Roll-Off measures the spectral concentration less than threshold.	$\lambda\sum_{i=1}^{N}\lvert Z[i]\rvert\lambda \simeq 0.85$ (frequency where 85 percent of the speech signal power resides)	(Thoman, 2009)
Spectral Centroid	Spectral Centroid is the average frequency of speech signal weighted by magnitude.	$\sum_{i=1}^{N}i\times\lvert Z[i]\rvert/\sum_{i=1}^{N}\lvert Z[i]\rvert$	(Thoman, 2009)
Spectral Flatness	Spectral Flatness shows whether the distribution is spike or smooth.	$\left(\prod_{i=1}^{N}\boldsymbol{Z}[i]\right)^{1/N}\left(\sum_{i=1}^{N}\boldsymbol{Z}[i]\right)^{-1}$	(Kadiri, Prasad, & Yegnanarayana, 2020)
Spectral Entropy	Spectral Entropy calculate the regularity of power spectrum of speech signal.	$1/\log\boldsymbol{N}\left(\sum_{i=1}^{N}\boldsymbol{P}(Z[i])\log\boldsymbol{P}(Z[i])\right)$	(Luque-Suárez, Camarena-Ibarrola, & Chávez, 2019, Shannon, 2001)
Spectral Contrast	Spectral Contrast represents the relative distribution of frequency instead of average frequency of speech signal.	$\log\left(\frac{1}{\alpha N}\sum_{i=1}^{\alpha N}s_{k,i}\right)-\log\left(\frac{1}{\alpha N}\sum_{i=1}^{\alpha N}s_{k,N-i+1}\right)$ where N is a number in kth - sub-band, k ∈ [1,6]and$\alpha = 0.02$	(Jiang, Lu, Zhang, Tao, & Cai, 2002)

Table 7
List of prosodic features.

Feature	Description	References
Fundamentalfrequency (F0)	F0 is the reciprocal of time interval between two consecutive glottal cycles.	(Ajmera et al., 2011; Sekkate et al., 2019; Tirumala et al., 2017)
Pitch	It is a perceptual property of the speech signal with physical characteristics denoted by the F0.	(Ajmera et al., 2011; Sekkate et al., 2019; Tirumala et al., 2017)
Intensity	Intensity is the measure of loudness or energy of a signal and related to amplitude square.	(Sekkate et al., 2019)

speech signal must be divided into short segments (20–30 ms) which are supposed to be stationary (Benesty, Sondhi, & Huang, 2008; Deller, Proakis, & Hansen, 2000).

The time window is advanced every 10 ms (Fig. 6) that enables the temporal features of each sound to be tracked and 25 ms analysis window is normally sufficient to extract discriminative spectral and temporal characteristics of these sounds. After framing, each individual overlapping frame is multiplied with Hamming or Hanning window (Picone, 1993) to smooth the edges and enhance the harmonics of each individual frame using Eq. 3.4.

- *DFT spectrum:* All the windowed frames are transformed into magnitude spectrum by using DFT.

$$X(k)=\sum_{n=0}^{N-1}x(n)e^{\frac{-j2\pi nk}{N}};0\le k\le N-1 \tag{4.2}$$

where N represents the number of samples used for DFT calculation.

- Mel spectrum: The Fourier transformed signal from step 2 is passed through a series of band-pass filters called as Mel-filter bank. Mel is a unit of measure which is based on the estimated human ears frequency. The estimation of Mel-scale can be written as

$$Mel(f)=2595^{*}log_{10}\left(1+\frac{f}{700}\right) \tag{4.3}$$

where f represents the physical frequency and $Mel(f)$ represents the estimated frequency.

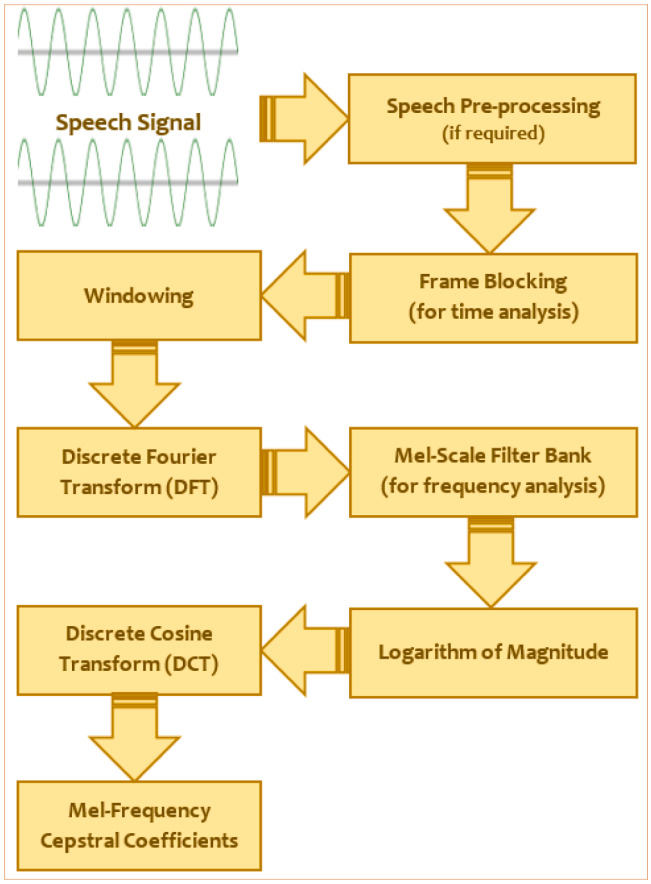

Fig. 7. MFCC feature extraction technique.

The Mel spectrum of $X(k)$ is calculated by multiplying the every triangular filter with magnitude spectrum using Eq. (4.4).

$$s(m)=\sum_{k=0}^{N-1}\left[\lvert X(k)\rvert^2\times H_m(k)\right];0\le m\le M-1 \tag{4.4}$$

where M represents the number of triangular filters. $H_m(k)$ is the weight assign to k^{th} bin of spectrum, which contributes to the m^{th} output band

and is written as:

$$H_m(k) = \begin{cases} 0, k < f(m-1) \\ \dfrac{2(k - f(m-1))}{f(m) - f(m-1)}, f(m-1) \le k \le f(m) \\ \dfrac{2(f(m+1) - k)}{f(m+1) - f(m)}, f(m) < k \le f(m+1) \\ 0, k > f(m+1) \end{cases} \quad (4.5)$$

with m varies from 0 to $M-1$

- *Discrete cosine transform:* Before computing DCT, Mel spectrum is measured on logarithmic scale and finally MFCC features are derived by computing the DCT of all log Mel spectrums (Eq. (4.6)).

$$c(n) = \sum_{m=0}^{M-1} \log_{10}\left(s(m)\cos\left(\frac{\pi n(m-0.5)}{M}\right)\right); n = 0, 1, 2, \cdots, C-1 \quad (4.6)$$

where $c(n)$ represents cepstral coefficients and C represents the total number of MFCCs. Generally, speaker identification systems use first 8 to 13 cepstral coefficients since these coefficients reflects the much of signal information which makes the system robust.

One of the methods for extracting discriminant features from human voice that best represent phonemes is Linear Prediction Coding (LPC). Linear Prediction Coding provide both time and frequency domain features using the correlation of all the adjacent samples and pole spectrum by means of resonance structure. In addition, LPC is efficient at providing an accurate approximation of speech spectra, formants and pitch using representation of natural human voice production system. Due to the simplicity and fast applicability, linear prediction coding is popularly utilized to extract and store time-varying formant in data in speaker identification. Another variant of LPC called Linear Predictive Cepstral Coefficient (LPCC) were recently proposed. LPCC is used to model the human voice production system in clean environment and applies a filter to simulate the vocal tract.

The thorough investigation of selected studies revealed that the short-term and voice source MFCCs features along with Delta and Delta–Delta coefficients were utilized in most studies as shown in Table 7. For instance, some researchers (Ali et al., 2018; Bisio, Garibotto, Grattarola, Lavagetto, & Sciarrone, 2018; Medikonda & Madasu, 2018) utilized MFCCs features for SI and reported that these features mainly consists of vocal tract information. It is calculated based on the filterbank method but implemented using the analysis of the time frequency. Firstly, the analysis of time is performed by employing framing process followed by analysis of frequency based on speech frame progression through filterbank. MFCC requires overlapping frames as time analysis is conducted in advance. Filterbanks are designed to operate in a way similar to human auditory perception of frequency (Ma & Leijon, 2011). Moreover, one of the reasons behind the extensive usage of the MFCCs features by the researchers is the high performance because mel-filter bands are positioned logarithmically and these features are more in line with human ear auditory characteristics (Sun et al., 2019). However, the performance of MFCC features degrade under noisy conditions (Hansen, Sarikaya, Yapanel, & Pellom, 2001). Furthermore, MFCC features concentrate on the whole spectral envelop of shorter frames and lack in speaker-discriminative features such as pitch information (Almaadeed et al., 2016; Nagrani et al., 2017). The other successful short-term based approaches utilized in reviewed studies include GFCC, LFCC, MFSC, RASTA, RASTA-PLP and CFCC. CFCC feature extraction approach is capable to address the acoustic disparity between training and test data. The features derived with CFCC are established better than the RASTA-PLP approach.

Features which best reflect phonemes can be extracted by understanding human auditory system. In this respect, LPC is one of the linear prediction-based approaches for extracting spectrum feature that offers good understanding for both frequency and time domain. LPC has the ability to provide a precise estimation of the speech formants, pitch and spectra by imitating the human auditory system. It is widely employed in SI due to its ability to extract and store time-varying formant data. For instance, Kawakami et al. (2014) reported that linear prediction features outperformed baseline MFCC features. However, fusion of linear prediction (LPC, LPCC) and short-term (MFCC) achieved best performance. The other linear prediction-based approaches used in reviewed studies such as PLP, EMD, DWT, WPT were also proven efficient.

Statistical features (mean, median, interquartile range, standard deviation) were derived in a research by Dhakal et al. (2019) and combined with features extracted from spectrogram and Gabor Filter for SI. The experimental results demonstrated that the combination of such features enhanced the accuracy of the prediction. Moreover, some studies (Lukic et al., 2016; Yadav & Rai, 2018) learned features from spectrogram and compared those features with conventional human-driven features. A spectrogram represents speech features such as pitch, energy, formants, time and fundamental frequency in the image form (He, Lech, Maddage, & Allen, 2009; He, Lech, Memon, & Allen, 2008). Furthermore, some studies (Mokgonyane, Sefara, Manamela, & Modipa, 2019) utilized spectral features (spectral centroid, spectral spread, spectral flux, spectral entropy, spectral roll-off and spectral density etc.) by converting the speech signal into frequency-domain using Fourier transform. Such features can be utilized to identify the pitch, notes, melody and rhythm.

Medikonda and Madasu (2018) proposed Type-2 Information Set (T2IS) and Hanman Transform (HT) features as higher order information set for text-independent SI system. Initially, MFCC features were extracted from each frame of speech signal in matrix form, where each row corresponds to single dimension and column corresponds to single frame. This MFCC matrix facilitated to derive T2IS and HT features by considering the temporal and cepstral possibilistic uncertainties. The proposed T2IS and HT features shown reduced computational time and complexity, feature size and also performance of these features was not degraded under noisy conditions.

The summary of features extracted in the reviewed studies is shown in Table 7.

4.2. Features extraction toolboxes

Features extraction is the main building block of modern speech processing research and development. Therefore, a number of speech feature extraction toolboxes and libraries have been developed that offer a broad range of speech-related functionalities such as workflow, preprocessing, algorithm efficiency and visualization problems (Moffat, Ronan, & Reiss, 2015). Despite significant research in the area of speech processing and features extraction, little research has been done on identifying and evaluating suitable features extraction toolboxes and their relevant applications. The feature extraction toolboxes that were utilized in reviewed studies for SI are:

Markov Model ToolKit (HTK) (Young & Young, 1993) is an open source freely available toolbox under the license of Microsoft Corporation for constructing hidden Markov models (HMMs). It is implemented in C programming language and runs mostly on UNIX-based systems while it can run on any modern operating system. HTK was primarily designed for speech recognition however it can be used for various other applications such as automatic speaker recognition, speech emotion recognition, DNA sequencing and character recognition.

SIDEKIT (Larcher, Lee, & Meignier, 2016) is an efficient open source toolbox that provide analysis of speech data for language and speaker identification systems. It has been designed and implemented in Python and distributed under LGPL license for both MacOS and Linux. SIDEKIT provides a simple interface for extracting and normalizing cepstral coefficients with Mel- or linear-scale filter bank (MFCC, LFCC). Moreover, GMM, Probabilistic Linear Discriminant Analysis (PLDA), i-vectors,

Table 8
Feature extraction methods used in the reviewed studies.

Feature extraction methods	Reference
MFCC + Spectrogram + log-mel Filterbank	An et al. (2019a)
MFCC + Spectrogram + MFSC	Imran et al. (2019)
MFCC + Spectrogram	Bunrit et al. (2019), Liu et al. (2018)
MFCC + Delta MFCC + Delta–Delta MFCC	Ali et al. (2018), Bisio et al. (2018), Dovydaitis and Rudžionis (2018), Soleymanpour and Marvi (2017)
MFCC + ZCR + Spectral Roll-off + Roughness + Brightness + Irregularity	Sardar and Shirbahadurkar (2018b)
MFCC + LFCC + LPC + ZCR + Spectral Roll-off + Brightness + Irregularity + Roughness	Sardar and Shirbahadurkar (2019)
MFCC + Delta MFCC + Delta–Delta MFCC + GFCC	Medikonda and Madasu (2018), Zhang et al. (2018)
MFCC + LPCC + DGS + DGCS	Sun et al. (2019)
MFCC + LPC + LBP	Abdul (2019)
MFCC + Spectral Roll-off + Brightness + Roughness + Irregularity	Sardar and Shirbahadurkar (2018a)
MFCC + ZCR + Spectral Centroid + Spectral Entropy + Spectral Flux + Spectral Spread + Spectral Roll-off + Energy + Entropy of Energy + Chroma Deviation + Chroma Vector	Mokgonyane et al. (2019)
MFCC + LPC	Almaadeed et al. (2016)
MFCC + DWT + WPT + WSBC	Almaadeed, Aggoun, and Amira (2015)
MFCC + LPCC + LPC residual + phase	Kawakami et al. (2014)
MFCC	Mporas et al. (2016), Tirumala and Shahamiri (2017)
MFCC + GFCC	Jawarkar et al. (2015), Zhao et al. (2014)
MFCC + LFCC	Fan and Hansen (2010)
MFCC + Spectral and Cepstrum periodicities	Al-Rawahy et al. (2012a, 2012b)
MFCC + Delta MFCC + Delta–Delta MFCC + Spectrogram	Chunlei Zhang, Koishida, and Hansen (2018a)
MFCC + CFCC + GFCC + RASTA + RASTA-PLP	Li and Gao (2016)
MFCC + Delta MFCC	Michalevsky et al. (2011)
MFCC + BFCC + PLP + RASTA-PLP	Abdalmalak and Gallardo-Antolín (2018)
MFCC + LPCC	Sadıç and Bilginer Gülmezoğlu (2011)
MFCC + IMMFC	Chakroborty and Saha (2009)
MKMFCCs	Faragallah (2018)
Hanman Transform (HT) + Type-2 Information Set (T2IS)	Medikonda and Madasu (2018)
Frame based linear predictive coding spectrum (FBLPCS)	Wu and Lin (2009a)
Spectrogram	Lukic et al. (2016), Yadav and Rai (2018)
Statistical Features (mean, median, interquartile range, standard deviation) + Gabor Filter + Spectrogram	Dhakal et al. (2019)
WPT + DWT	Wu and Lin (2009b)
LPC + DWT	Sarma and Sarma (2013)
Spectrogram + Radon transform	Ajmera et al. (2011)
Vocal resonant frequencies (F1, F2, F3) + Wavelet Packet Transform (WPT)	Daqrouq (2011)
Empirical mode decomposition (EMD)	Wu and Tsai (2011)
PWCC + PWPCC + SPWPCC	Renisha and Jayasree (2019)
Coiflet Wavelet	Indumathi and Chandra (2015)
Statistical Features (Min, Max, Mean, Median, Mode, Variance, Standard Deviation) + ZCR + RMS	Jahangir, Teh, Ishtiaq, Mujtaba, and Nweke (2018)
Pitch + intonation + timing + loudness	Manikandan and Chandra (2016)
Vocal resonant frequencies (F1, F2, F3, F4, F5) + Wavelet Packet Transform (WPT)	Daqrouq and Tutunji (2015)

Joint Factor Analysis, DNN and SVM are also implemented for modeling and classification. Finally, SIDEKIT provides tools to calculate Decision Cost Function, minimum Decision Cost Function and Equal Error Rate and plot two types of DET curves: ROC Convex-Hull and steppy (from ROC).

openSMILE (Eyben, Weninger, Gross, & Schuller, 2013) developed by audEERING for academic and scientific purposes, is an open source feature extraction library for music and speech applications. In addition, it is a platform independent library written in C++, which associates the features of speech processing and music information retrieval. It is compatible with the data-formats of LibSVM, HTK and WEKA, as it supports the LLDs described in Table 8.

Kaldi (Povey et al., 2011) is publicly available and most commonly used feature extraction library for speaker and speech recognition tasks. Its code, written in C++, is extendable and modifiable. Additionally, it supports Windows operating system and different versions of UNIX. Kaldi intends to extract MFCC and PLP features. Moreover, it also supports VTLN, LDA, HLDA, and STC/MLLT, variance normalization, cepstral mean and other feature extraction approaches.

Python_Speech_Features (PSF) is publicly available toolkit, developed by James Lyons for the researchers' community working in the discipline of speech, speaker and emotion recognition. Moreover, the features supported in this toolkit are MFCC, Log FbE, Filterbank Energies (FbE) and Spectral Subband Centroids (SSC). The default parameter setting is provided for each mentioned speech feature but the parameters of features can be modified in accordance with the area under consideration.

jAudio (McKay, Fujinaga, & Depalle, 2005) is implemented in Java in order to exploit on cross-platform portability and development advantages of Java. It provides GUI where user give audio files as input and select acoustic features (Zero Crossing, RMS, Low Amplitude Frames, Spectral Flux, Spectral Rolloff, Compactness, Method of Moments, MFCC and Beat Histogram) that they want to extract and execute directly from GUI. The results can be stored in an ARFF file or an XML file depending on individual's choice. Additionally, it also provides a command-line interface in order to extract features via scripting.

YAAFE (Mathieu, Essid, Fillon, Prado, & Richard, 2010) YAAFE, developed in Python, is a command-line toolbox. The user needs to provide a feature extraction plan along with audio files in a text file where each line defines the feature name, parameters and transformations (name: Feature param = value param = value). Firstly, a parser examines the feature extraction plan in order to find the common computational measures, and a reduced data flow graph is generated. Secondly, feature extraction is applied to the provided audio files based on the reduced data flow graph and output is stored in the HDF5 files. The main strength of YAAFE is its considerably less complexity because of the correct exploitation of redundancy in the calculation of features. In addition, YAAFE is easy to configure and all features can be parametrized independently.

LibXtract (Bullock & Conservatoire, 2007) is a cross-platform, open-

source and free library that includes a superset of audio features. The library is implemented in ANSI C and licensed under GNU General Public License (GPL) in order to incorporate with any software that supports linking to shared libraries. LibXtract divides audio features into vector features, scalar features and delta features which return the output as an array, single value and temporal element respectively. Moreover, some of the audio features provided by the library are Vector Mean, Vector Kurtosis, Spectral Mean, Spectral Kurtosis, Spectral Centroid, Spectral Smoothness, Spectral Spread, Vector Zero Cross Rate, Signal Loudness, Spectral Inharmonicity, Signal Fundamental Frequency and MFCC. Additionally, LibXtract provide vamp-LibXtract-plugin that make the entire set of features available to a vamp host.

Essentia (Bogdanov et al., 2013) published under the Affero license, is an open-source and cross-platform C++ toolkit for audio analysis. It is comprised of a collection of reusable algorithms for extracting features from audio. The algorithms include input/output functionality of audio file, digital signal processing building blocks, general algorithms for numerical characterization, and temporal, tonal, spectral and high-level music descriptors.

pyAudioAnalysis (Giannakopoulos, 2015), licensed under Apache License, is an open source toolkit for different audio signal processes, including feature extraction, regression, unsupervised and supervised segmentation and classification of auditory signals. The applications of pyAudioAnalysis include, audio event detection, music segmentation, movie recommendation, SI, depression classification and healthcare. Moreover, the extracted features are categorized into frequency-domain, time-domain and cepstral-domain features.

Although the provided list (Table 8) is not extensive, but the recent, popular and programming environment related toolboxes have been analyzed.

4.3. Feature selection/reduction methods

The -
whole -
set of features derived from the database includes irrelevant features that may reduce the accuracy during classification step. Thus, feature reduction is another important phase in feature engineering which is a method for discovering a subset of input features that are most suitable to build a classification model. The goal is to simplify the classification models, reduce computational complexity, reduce input dimensionality and avoid overfitting. The feature reduction methods (Table 9) that were utilized on the selected primary studies are Principle Component Analysis (PCA), Sequential Backward Generation (SBG) and Sequential

Table 9
Features extractor toolboxes used in SI.

	HTK	SIDEKIT	Open SMILE	Kaldi	pyAA	jAudio	YAAFE	Essentia	LibXtract	Librosa
Organization	Cambridge University	LIUM, Universite du Mans	audEERING GmbH	Microsoft Research	Behavioral Signals	QMUL	Telecom ParisTech	Pompeu Fabra University	–	NYU
Platform	Windows LINUX	Windows MacOS LINUX	Windows MacOS LINUX	Windows MacOS LINUX	Windows MacOS LINUX	Windows MacOS LINUX	MacOS LINUX	Windows MacOS LINUX	Cross-Platform	Cross-Platform
Open Source	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Access	Free[a]	Free[b]	Free[c]	Free[d]	Free[e]	Free[f]	Free[g]	Free[h]	Free[i]	Free[j]
Language Support	Python MATLAB C	Python	C++	C++	Python	Java	Python MATLAB C++	Python C++	C Java	Python
MFCC	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
LPC	✓	×	✓	×	×	×	✓	✓	×	×
LPCC	×	×	×	×	×	×	×	×	×	×
LSF	×	×	×	×	×	×	✓	×	×	×
PLP	✓	×	✓	✓	×	×	×	×	×	×
LFCC	×	✓	×	×	×	×	×	×	×	×
ZCR	×	×	✓	×	✓	✓	✓	✓	✓	✓
HNR	×	×	✓	×	×	×	×	✓	×	×
RMS	×	×	×	×	×	✓	×	×	×	✓
MoM	×	×	×	×	×	✓	×	×	×	×
Energy	×	×	✓	×	✓	×	✓	✓	×	×
Formant	×	×	✓	×	×	×	×	×	✓	×
Tonal	×	×	✓	×	×	×	×	✓	×	✓
Spectral Flux	×	×	×	×	✓	✓	✓	✓	×	✓
Spectral ROF	×	×	×	×	✓	✓	✓	✓	×	✓
Spectral BW	×	×	×	×	×	×	×	×	×	✓
Mean	×	×	×	×	×	×	×	×	✓	×
Kurtosis	×	×	×	×	×	×	×	×	✓	×
S. Centroid	×	×	×	×	✓	×	×	✓	✓	✓
Spectrogram	×	×	×	×	✓	×	×	×	×	✓
Chroma	×	×	×	×	✓	×	×	×	×	✓
Output	HTK	SPRO4 HTK	Matrix CSV HTK ARFF	Matrix	CSV Matrix	XML ARFF	CSV HDF5	YAML JSON	VAMP XML	CSV TSV

**MoM = Method of Moments, Spectral ROF = Spectral Rolloff, S. Centr = Spectral Centroid, Spectral BW = Spectral Bandwidth, pyAA = pyAudioAnalysis, QMUL = Queen Mary University of London, NYU = New York University.

[a] http://htk.eng.cam.ac.uk/download.shtml.
[b] pip install sidekit.
[c] https://www.audeering.com/opensmile/.
[d] https://sourceforge.net/projects/kaldi/.
[e] https://github.com/tyiannak/pyAudioAnalysis/.
[f] https://sourceforge.net/projects/jaudio/files/.
[g] http://yaafe.sourceforge.net/.
[h] https://essentia.upf.edu/download.html.
[i] https://sourceforge.net/projects/libxtract/.
[j] https://github.com/librosa/librosa.

Forward Generation (SFG) which are briefly described below.

4.3.1. Principle component analysis

Principle Component Analysis is widely employed to reduce the dimensionality of input data, while retaining the utmost 'variability'. Most of the selected primary studies employed PCA to reduce the dimensions of feature set by eliminating the redundant or selecting only those features which contribute to the classifier training. The general process (Ahmad, Thosar, Nirmal, & Pande, 2015) for estimating PCA is as follows

1. Calculate the x_{mean} of the given feature vector $\{x_1, x_2, x_3, \cdots, x_n\}$ as

$$x_{mean} = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{4.1}$$

2. Compute covariance matrix *Cov* as

$$Cov = \frac{1}{n}\sum_{i=1}^{n} \bar{x}_i.\bar{x}_i^T \tag{4.2}$$

where $\bar{x}_i = x_i - x_{mean}$

3. Lastly, calculate the eigenvalues and eigenvectors of the corresponding *Cov* using Eq (4.3).

$$Cov \times e_k = \lambda_k \times e_k for k = 0, 1, \cdots n-1 \tag{4.3}$$

Where $e_k = k^{th} eigenvector, \lambda_k = k^{th} eigenvalue$.

4. In order to minimize the dimensions of feature vectors, eigenvectors corresponding to highest eigenvalues are chosen (Shlens, 2014).

$$M_{fv} = (e_1, e_2, e_3, \cdots, e_p) \tag{4.4}$$

where $p = number of dimensions of PCA matrix$.

5. The new feature vector is computed as

$$N_{fv} = M'_{fv} \times T \tag{4.5}$$

where $T = transpose of the mean adjusted original input$.

Recently, Ali et al. (2018) employed PCA to reduce the dimensions of audio scripts by linearly transforming the frequencies to low-dimensional space. The authors preserve just 80 components out of total 256 dimensions after applying PCA.

4.3.2. Sequential forward generation

Sequential Forward Generation starts with an empty feature set, S_{select}. As search begins, best feature is selected among unselected ones based on certain criteria at a time (hence, sequential) and added into S_{select}. This process repeats until S_{select} achieves a complete set of novel features. An ordered list of the selected features can also be obtained depending on how early a feature is added to the list and then first m relevant features are chosen from the list (Liu & Motoda, 2012; Sardar & Shirbahadurkar, 2018a).

4.3.3. Sequential backward generation

Sequential Backward Generation is a backward counterpart of SFG. It starts with a full feature set, S_{select}. As search begins, least important feature is selected from feature set based on certain criteria and removed from S_{select}. So, the S_{select} reduces until there is just one feature. An ordered list of the selected features can also be obtained depending on how late a feature is removed from the list and then last m relevant features are chosen from the list (Sardar & Shirbahadurkar, 2018a).

Table 10
. Feature selection methods used in reviewed studies.

Methods	Advantage	Disadvantage	Reference
PCA	reduce the number of dimensions without losing information	not scale-invariant	Ali et al. (2018), Indumathi and Chandra (2015), Sardar and Shirbahadurkar (2018a), Zhang et al. (2018a), Zhang et al. (2015)
SFG	best performs when the optimum feature subset is small	unable to subtract features that become redundant after adding other features	Sardar and Shirbahadurkar (2018a)
SBG	best perform when the optimum feature subset is big, as it spends most of time exploring large subsets	unable to re-evaluate the usefulness of a feature after its removal	Sardar and Shirbahadurkar (2018a)
UFSM	simple to understand, run and provides a better data understanding	Not always give optimized feature set	Dhakal et al. (2019)

4.3.4. Univariate feature selection method

Univariate feature selection method (UFSM) selects the best features on the basis of univariate statistical tests. It examines every feature individually to evaluate the strength of the feature relationship with response variable.

5. Machine learning methods

At this stage, a classification model is constructed on the training data by employing the ML algorithm. The constructed model has the capability to predict the unknown utterances of speaker. Various machine learning algorithms (Table 10) have been implemented for automatic speaker identification. These machine learning algorithms include Gaussian Mixture Model, Support Vector Machine, Naïve Bayes, Decision Tree and Artificial Neural Network. These algorithms are presented in the following subsection.Table 11.

5.1. Gaussian Mixture model

A GMM (Reynolds & Rose, 1995) can be interpreted as a multivariate probability distribution method which is suitable for arbitrary distributions modeling and currently it is one of the most widely used methods for SI systems. The GMM of feature vectors distribution for speaker s is a weighted linear mixture of M unimodal Gaussian densities $b_i^s(x)$, which are parameterized through the mean vectors μ_i^s with a covariance matrix $\sum_i^s$. These parameters, which jointly construct a speaker model, are denoted by the notation $\{p_i^s, \mu_i^s \sum_i^s\}_{i=1}^M$. The p_i^s represents the mixture weights which satisfy the stochastic constraint $\sum_{i=1}^M p_i^s = 1$. For a feature vector x, the density mixture for speakers is determined as

$$p(x|\lambda_s) = \sum_{i=1}^{M} p_i^s b_i^s(x) \tag{5.1}$$

where,

$$b_i^s(x) = \frac{1}{(2\pi)^{D/2}|\sum_i^s|^{1/2}} e^{\left(-\frac{1}{2}(x-\mu_i^s)'(\sum_i^s)^{-1}(x-\mu_i^s)\right)} \tag{5.2}$$

and D is the feature vector dimension.

Table 11
Machine Learning Classification algorithm used in reviewed studies.

Method	Strengths	Weaknesses	Reference
GMM	It needs a smaller number of parameters for learning which can be accurately estimated by adopting the expectation maximization.	It needs sufficient data to model the speaker well.	Al-Rawahy et al. (2012a, 2012b), Chakroborty and Saha (2009), Fan and Hansen (2010), Jawarkar et al. (2015), Kawakami et al. (2014), Li and Gao (2016), Michalevsky et al. (2011), Sadıç and Bilginer Gülmezoğlu (2011), Wu and Lin (2009a), Zhao et al. (2014)
DT	Ignores irrelevant features, very fast for predicting unknown records and easy to construct.	Slight variations in training data may produce large variations to decision logic, Easy to overfit and large decision trees can be challenging to interpret.	Dhakal et al. (2019), Indumathi and Chandra (2015), Jahangir et al. (2018), Manikandan and Chandra (2016), Mporas et al. (2016)
NB	Needs less amount of training data to approximate the parameters required for prediction.	NB cannot learn the relationship between the features.	Indumathi and Chandra (2015), Jahangir et al. (2018)
SVM	It is memory efficient and perform well when target classes are not overlapping.	Not suitable for massive databases and under-perform when number of features are more than number of training examples.	Abdalmalak and Gallardo-Antolín (2018), Abdul (2019), Dhakal et al. (2019), Jahangir et al. (2018), Medikonda and Madasu (2018), Mokgonyane et al. (2019), Mporas et al. (2016), Wang et al. (2015)
k-NN	No training data is required prior to making predictions. Therefore, *k*-NN classifier much faster than other classifiers that need training data such as SVM.	*k*-NN is not suitable for high dimensional and categorical features as it is challenging to find the distance in every dimension. Moreover, *k*-NN has high classification cost for massive datasets.	Abdul (2019), Ajmera et al. (2011), Jahangir et al. (2018), Medikonda and Madasu (2018), Michalevsky et al. (2011), Mokgonyane et al. (2019), Mporas et al. (2016), Sardar and Shirbahadurkar (2018a), Sardar and Shirbahadurkar (2018b), Sardar and Shirbahadurkar (2019)
+	It provides robust and efficient techniques to learn feature representation automatically from complex speech data.	It requires huge training data, can stuck at the local optima and challenging to construct explicit model.	Almaadeed et al. (2015), Daqrouq (2011), Daqrouq and Tutunji (2015), Krothapalli, Yadav, Sarkar, Koolagudi, and Vuppala (2012), Mporas et al. (2016), Renisha and Jayasree (2019), Sarma and Sarma (2013), Soleymanpour and Marvi (2017), Wu and Lin (2009a), Wu and Lin (2009b), Wu and Tsai (2011)

Given a feature vectors $X = \{x_1, x_2, \cdots, x_T\}$ for a speech signal with T frames, the log-probability for speaker model s is computed as

$$L_s(X) = \log p(X|\lambda_s) = \sum_{t=1}^{T} \log p(x_t|\lambda_s) \tag{5.3}$$

For SI, the value of $L_s(X)$ is calculated for each enrolled speaker modelλ_s and the model with maximum likelihood is returned as identified speaker. During model training, feature vectors are trained using the Expectation Maximization (EM) (Covoes & Hruschka, 2013; Tian, Xia, Zhang, & Feng, 2011) algorithm. This algorithm involves an iterative process to update each parameter in λ, with a consequential increase in the log-probability at every step.

From the review studies, a number of studies have implemented Gaussian Mixture Model (GMM) for speaker identification. For instance, Kawakami et al. (2014) conducted SI experiments for JNAS database that contains 135 females and 135 males, about 100 utterances per speaker. Each speaker GMM model was trained through five utterances for four features (MFCC, LPC residual, phase, LPCC). The rest of the utterances were used for the test. In this study, for test data, the authors used short utterances by cutting complete utterances into 2, 0.5 and 1 s, as well as the complete one. The experiments result of single vocal tract features showed that the LPCC achieved better performance than the MFCC, and from single vocal source features, LPC residual performed better than phase. Nevertheless, the combination of both vocal tract and vocal source features achieved highest accuracy of 98.4% using whole utterances.

In another study, Jawarkar et al. (2015) investigated the influence of two non-linear compression functions: cubic-root and log used in GFCC and MFCC features extraction, on the performance of SI system under clean and noisy conditions. The GMM approach was employed for speaker modeling using Hindi and Marathi speech databases. The authors observed that the cepstral features based on cubic-root achieved better performance than the cepstral features based on log under noisy conditions having SNR $<$ 20 dB. However, log-based MFCC outperformed GFCC for test data with SNR $>=$20 dB.

There are various known acoustic features such as MFCCs which is most successful for SI systems. Additionally, Al-Rawahy et al. (2012a, 2012b), introduced a new set of features, namely, DCT-Cepstrum Coefficients Histogram, inspired by MFCCs common usage, but faster in computation. The authors implemented a text-independent SI system based on DCT-Cepstrum Coefficients Histogram and GMM. The proposed model was evaluated using audio files from ELSDSR and TIMIT databases, and achieved high classification accuracy of 99% on TIMIT corpus and 100% on ELSDSR database.

In their recent study, Sadıç and Bilginer Gülmezoğlu (2011) examined the performance of proposed CVA–GMM and other approaches including FLDA and GMM using TIMIT database. In this study, 20-dimensional MFCC feature vector was extracted from each frame with 256 samples. The achieved recognition rates indicate that GMM and CVA performed better than FLDA. However, FLDA obtained good results in terms of memory requirement and processing time as compared to GMM. Moreover, GMM with 32 mixtures (GMM32) provided slightly better classification results than those achieved from GMM with 16 mixtures (GMM16). However, the memory requirement and processing time of GMM32 were much higher than GMM16. Furthermore, the classification results achieved from GMM and CVA were similar but CVA outperformed GMM in terms of memory requirement and processing power. In order to achieve better classification results, the authors proposed a new CVA–GMM method.

Over the years, MFCC has been utilized as a regular acoustic feature set for various speech related applications. In a study by Chakroborty and Saha (2009), authors examine the benefit of IMFCC feature set for SI, which provides additional information present in the high frequency region. This study introduced the Gaussian filter (GF) while computing MFCC and IMFCC instead of regular triangular filters. GMM was employed for speaker modeling and the performance of both MFCC and IMFCC improved with GF over regular triangular filter in two benchmark databases YOHO and POLYCOST.

Furthermore, Li and Gao (2016) performed voice activation detection (VAD) (Nemer, Goubran, & Mahmoud, 2001) on the raw speech signals before obtaining the feature vectors. This study used a 26-dimensional MFCCs, a 40-dimensional CFCCs, a 48-dimensional GFCCs and a

34-dimensional RASTA-PLP. Under clean environment, GMM and GMM-UBM models were trained on each type of acoustic feature and experimental results showed that the proposed GFCC features outperformed MFCC, CFCC and RASTA-PLP in both models. Compared with GMM, GMM–UBM model achieved higher recognition rate. In second experiment, the robustness of proposed GFCC feature was evaluated under four types (F-16 cockpit noise, tank noise, babble noise and white noise) of noisy conditions. It was found that the proposed GFCC feature achieved high classification accuracy in each type of noisy condition.

5.2. Decision tree

Decision Tree (DT) is a machine learning classifier that recursively divides training dataset into node segments consisting of root node, inner split and leaves nodes (Quinlan, 1986). Splitting of data is implemented on all nodes on the basis of simple feature with define stopping criteria. DT is non-parametric algorithm and doesn't need supposition on the partition of training data. In addition, DT can model the non-linear relationship between features and target classes (Friedl & Brodley, 1997). One of the benefits of DT is that, it does not need data preparation. However, DTs can become unstable as a small data variation can generate an entirely different tree. Verities of DT algorithms have been implemented in SI systems such as Random Forest (RF), J48, ID3 and C4.5.

Manikandan and Chandra (2016) investigated fuzzy based hierarchical DT approach for speaker identification. The key idea behind this approach was to get speaker clusters with improved efficiency using prosody features (intonation, loudness and timing) with fuzzy clustering at all levels to build the hierarchical DT. The reported result demonstrated that the proposed method using prosody features outperformed the speaker accuracy of 93.7% when compared to vocal features accuracy of 81.2% under noisy conditions.

Also, Jahangir et al. (2018) extracted statistical features (min, max, mean, mode, median, standard deviation, variance and covariance) together with zero-crossing rate and RMS from speech data collected from 5 females and 5 male speakers. These features were used to evaluate the performance of various classification algorithms including SVM, k-NN, RF, NB and J48. The authors proposed hierarchical classification approach where speaker gender was identification at first level and speaker was classified at second level. RF algorithm outperformed other four algorithms by achieving accuracies of 96.9%, 78% and 88.7% for gender, male and female classification models respectively.

5.3. Support vector machine

SVM is a non-linear and linear ML classifier that is widely employed as for pattern recognition applications. The SVM constructs a binary class classifier by using a convex optimization theory during training of model. SVM was first implemented by Cortes and Vapnik (1995) and utilizes optimum hyperplane that expands a decision boundary between the two classes of the target (Ralph Abbey & Wang, 2017). However, to solve multinomial classification tasks, SVM employs structural risk minimization model (Li et al., 2015) shown below.

$$\min_{w,\xi_k} J_1(w,\xi_k) = \frac{1}{2} w^{\mathcal{T}} w + c \sum_{i=1}^{N} \xi_k \tag{5.4}$$

The linear SVM (Suykens & Vandewalle, 1999) method projects an input feature vector x into $f(x)$ as follows:

$$f(x) = sign\left[\sum_{\iota=1}^{N} \alpha_\iota k(x; x_\iota) + \beta\right] \tag{5.5}$$

Where $x_\iota$are support vectors, β is a bias and α_ι are the weights and N represents the total support vectors.

SVM have been widely employed as primary classification algorithm for constructing speaker identification system. For instance, Abdalmalak and Gallardo-Antolín (2018) extracted MFCCs, ΔMFCCs, ΔΔMFCCs, PLP, BFCC and RASTA-PLP features to compare the performance of 13 different combinations of extracted acoustic features. Finally, classified the speech signals using the combination of three different classifiers (linear kernel SVM, RBF kernel SVM and linear regression) in order to enhance the generalization and learning capabilities of the one classifier. It was observed that MFCC + PLP + RASTA-PLP + BFCC give the best performance with 98% accuracy using clean speech.

Each audio file in the study proposed by Mporas et al. (2016) was initially pre-processed by applying the energy-based voice activity detector (VAD) to retain only spoken parts. The speech signals were divided into frames using Hamming window of 20 ms with 10 ms time shift between consecutive frames. From each frame, 19 MFCCCs were calculated, which were expanded to delta and delta–delta coefficients and then CMVN and RASTA processing were applied in order to make these features more robust. For classification stage, SVM (radial based kernel), MLP, k-NN (IBk) and DT (C4.5) classification algorithms were employed. For the RSR2015 database, SVM obtained best results with the accuracies of 96.2% and 88.4% for text-dependent and text-independent SI.

Furthermore, Wang et al. (2015) used Gabor filters for signals filtering and then empirical mode decomposition (EMD) were transformed by employing Hilbert transform. The authors constructed the probability distribution of instantaneous frequencies, comparing non-parametric and parametric probability density modelling efficiency. Finally, the SVM that uses Riemann sum to estimate the probability product kernel was employed for classification. The experimental results indicated that non-parametric modeling outperformed parametric modeling by 15% for EMD-based system and by 17% for Gabor filter-based system.

In a recent study, Faragallah (2018) classified the speakers in Arabic database using support vector machine. The MKMFCCs features were extracted from speech signals for training of SVM classifier. The proposed MKMFCC–SVM outperformed MFCC–SVM under noisy conditions. In another study, Sun et al. (2019) extracted characteristics of speakers personality by introducing deep Gaussian correlation supervector features (DGCS) using a hybrid DBN-GMM model. The proposed DGCS features were fed as the input to SVM and proposed feature obtain 98% accuracy as compared to traditional i-vectors 97% and correlation supervector 96%.

5.4. k-Nearest neighbor

k-NN is a fundamental ML classifier that does not need the prior knowledge about the data. The key parameters used in *k-NN* are a distance function (Euclidean or Manhattan), nearest neighbors (k), decision rule and number of labeled samples of speech signals *Xn*. The test feature vector is assigned a class label based on the nearest distance from the existing training classes. Mathematically, a posteriori probability $P(\omega_i|x)$ of class is calculated as:

$$P(\omega_i|x) = \frac{k_i}{k} \times P(\omega_i) \tag{5.6}$$

where k_i represents the number of vectors belonging to class ω_i in a subset of k vectors (Shah, Smolenski, Yantorno, & Iyer, 2004). In general, a high value of k is suggested to minimize the influence of noise on classifier performance. Moreover, the odd value of k is selected for the binary classification. The results of *k-NN* classifier are also influenced by the manner in which distance between training feature vectors and testing feature vectors are computed by the different distance metrics.

5.4.1. Distance metric

The *k-NN* classifier predicts a class label of test speech signal based on the minimum distance from training classes called nearest neighbor.

A distance between coordinates of acoustic feature-vector of existing training classes (p) and coordinates of test feature vector (q) is computed. The Euclidean distance is computed as:

$$D(p,q) = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2} \tag{5.7}$$

Similarly, Manhattan distance between a pair of n-dimensional points (p,q) is computed as:

$$\sum_{i=1}^{n} |p_i - q_i| \tag{5.8}$$

The graphical representation of the Euclidean and Manhattan distance is shown in Fig. 7. Vector comprise of various acoustic features; some of the acoustic features may have strong intra-speaker differences (though not desirable) for some of the speech samples. The influence of such a strong difference is reduced because the distances are not squared in Manhattan distance.

k-NN is simple classifier which is useful for interpreting performance, accurate, fast as compared to other supervised ML algorithms. Due to the performance of k-NN, various studies have proposed the algorithm for speaker identification. For instance, Medikonda and Madasu (2018) proposed T2IS and HT features for text-independent SI. The MFCC features of speech signals were transformed into T2IS and HT features by considering the temporal and cepstral possibilistic uncertainties. These features were classified by SVM, k-NN and Improved Hanman Classifier (IHC). The performance of proposed methods was evaluated in terms of accuracy, computational complexity, speed and memory requirement on three databases namely VCTK, NIST2003 and VoxForge. The proposed features using k-NN classifier outperformed baseline GFCC, MFCC, ΔMFCC and ΔΔMFCC features under noisy environment at different SNRs.

Recent studies by Sardar and Shirbahadurkar (2018b), Sardar and Shirbahadurkar (2018a) and Sardar and Shirbahadurkar (2019) presented the timbrel features selected from whispered and neutral speech through hybrid selection approach. The combination of various timbrel features namely ZCR, spectral roll-off, brightness, irregularity, roughness and MFCCs were compared in three different speech modes i.e. whisper–whisper, neutral–neutral and neutral–whisper. However, authors targeted neutral–whispered mode of speech for SI. In addition, the effect of distance methods used in k-NN on the recognition accuracy was also investigated. The result of various experiments showed that timbrel features achieved 6% increase in accuracy compared to MFCCs with neutral–whisper mode. In the second experiment, accuracy was observed 6% more by using City-block (Manhattan) compared to Euclidean at similar environments. The combination of various timbrel features and k-NN classifier with City-block distance measure achieved the highest accuracy.

In pattern recognition applications, the features extracted from input data should have lowest intra-class variation while the inter-class variance should be maximum. Ajmera et al. (2011) proposed a new feature extraction method for SI using discrete cosine transform (DCT) and Radon transform (RT). In the proposed model, the effective acoustic features were extracted from speech spectrograms by applying RT technique which applies the values of pixel in the given spectrogram with a straight line in a specific direction and displacement. Further, the proposed method computed Radon projections in different orientations and captured the acoustic characteristics of spectrogram. DCT was applied on the Radon projections in order to get a low-dimensional feature vector. In the classification process, the feature vector derived from unknown utterance was compared with the feature vectors stored in the database using k-NN with Euclidean distance to make the final prediction. The proposed model, evaluated on TIMIT database, achieved 96.7% and 98.4% accuracy respectively on authors own created SGGS database.

5.5. Naïve Bayes

Naïve Bayes is widely used ML classifier build on Bays theorem that operates on the assumption that every data pair is independent and is equivalent in calculation of predictive feature. This theorem depends on the probability calculation of an event with respect to the probability of an event that has already occurred. Mathematically, this can be written as:

$$P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)} \tag{5.9}$$

Given an acoustic features set $\{x_1, x_2, x_3, x_4, \cdots x_n\}$ and class set $\{s_1, s_2, s_3, \cdots s_k\}$, the probability of features occurring in every class to find the most likely Speaker ID is computed as:

$$P(s_i|x_1 \cdots x_n) = \frac{P(x_1|s_i)P(x_2|s_i)\cdots P(x_n|s_i)P(s_i)}{P(x_1)P(x_2)\cdots P(x_n)} \tag{5.10}$$

Since the denominator is a constant value for all input values, therefore, it can be removed.

$$P(s_i|x_1 \cdots x_n) \propto P(s_i) \prod_{j=1}^{n} P(x_j|s_i) \tag{5.11}$$

The probabilities of all features for "c_i" class and return the result with maximum probability as;

$$\mathrm{y} = \underset{s_i}{\operatorname{argmax}} P(s_i) \prod_{j=1}^{n} P(x_j|s_i) \tag{5.12}$$

Naïve Bayes have also played prominent role in speaker identification using various feature vectors. In their recent study, Indumathi and Chandra (2015) extracted features from 50 samples by using Coiflet wavelets followed by feature selection through singular value decomposition for efficient Speaker Identification. The classification of samples was performed using NB, J48 and REPTREE classifiers. REPTREE and SVM provide the best performance with accuracies 94.3% and 91.4%, respectively. Additionally, Jahangir et al. (2018) evaluated the impacted of NB along with SVM, k-NN, RF and J48 classifiers to classify the extracted statistical features from speech data collected from 10 speakers. The experimental results showed that NB performed considerably with RF, K-NN, J48 and SVM.

5.6. Artificial neural network

Artificial Neural Network (ANN) typically comprises of one input layer, one output layer, and one hidden layer. Each layer consists of neuron-like information processing units, which are capable to learn automatically on the base of experience and estimating non-linear groupings of features for classification. ANN inputs are transferred through several layers to calculate the output of neuron by using weights and activation function and then updated through backpropagation to reduce error rate. The ANN provides robust and efficient techniques to learn representation of feature from complex speech data. However, ANN require huge training data, can stuck at the local optima (Hwang, Park, & Chang, 2016) and challenging to construct explicit classification model.

A study by Almaadeed et al. (2015) designed and implemented a text-independent SI system using wavelet transform analysis and ANNs in order to enhance the classification speed and accuracy. Initially, wavelet transform (Mallat, 1999; Vetterli & Kovačevic, 1995) was applied to decompose the given speech signal into a group of smaller signals at several levels and analyzed each component of the speech signal at various frequencies with various resolutions. Secondly, discriminative features were extracted from the entire speech signal. The methods used for feature extraction include WSBC, WPT, DWT and MFCC. The proposed approach combined multiple neural networks (MNNs) for the construction of SI model. The evaluation on GRID speech

database showed that proposed model outperformed classical GMM, PCA and BPNN in both identification accuracy and time.

Furthermore, Soleymanpour and Marvi (2017) proposed a novel approach to identify MFCC feature vectors with maximum similarity which is used to construct SI model and to define decision boundary. MFCC features were extracted from each frame of speech signal as a feature vector and then k-means clustering was used to get feature vectors with maximum similarity. The experiments were performed using ELSDSR speech database and ANN was employed as a classifier. Experimental results showed that the performance of SI system was improved in accuracy.

As vowel sounds occur more frequently in speech with higher energy, therefore, vowel phonemes can be utilized to extract discriminative features in noisy conditions. Sarma and Sarma (2013) presented a novel method for SI using the segmentation of vowel sound from words uttered by a speaker. The segmentation of vowel sound was performed through the combination of probabilistic neural network (PNN) and self-organizing map (SOM). Later, the segmented vowel sound was used for SI by matching patterns with an LVQ-based code book which was prepared by capturing features of vowel phonemes spoken by the speakers. The proposed SOM-based approach achieved 7% increase in accuracy compared with the DWT-based approach.

Moreover, Daqrouq and Tutunji (2015) proposed a new speaker identification based feature extraction method using wavelet entropy, formants and neural networks. Initially, seven Shannon wavelet entropy packets and five formants were extracted from speakers' signals as feature vector. In contrast to traditional SI methods that derives features from words (or sentences), the proposed method derived features from vowels. Secondly, these 12 extracted features coefficients were fed as input to feed-forward neural network. Using only 12 features coefficients, the proposed method achieved 89.16% accuracy and outperformed both MFCC-ANN and LPC-ANN.

6. Deep learning methods for speaker identification

Deep Learning(DL) is a powerful ML method with huge impact in several domains including image recognition, natural language processing, predictive forecasting, self-driving cars and computer vision, sensor data analysis and human activity recognition. DL techniques have achieved promising results in SI and speech recognition over traditional ML techniques such as MFCC features with GMMs. These DL techniques can be classified into two categories namely; discriminative and generative. The discriminative DL techniques (CNN, RNN) are modeled in a bottom-up method where data flows from input layer to output layer via hidden layers. These techniques are employed in supervised training for regression and classification problems. On the other hand, generative DL techniques (RBMs, DAE) use top-down approach where data flows in opposite way and these models are employed in unsupervised training (Shrestha & Mahmood, 2019). Given the input data x and corresponding class label y, discriminative DL techniques learn conditional probability distribution p ($y \mid x$), which is the probability of class label y given data x, whereas generative DL techniques learn the joint statistical probability distribution p (x, y), which can predict p (y|x) (Ng & Jordan, 2002). In general, when class label data is known (supervised training) discriminative techniques are employed as they offer effective training and generative techniques can be employed when class labeled data is not known (unsupervised training) (Bernardo et al., 2007). A number of deep learning techniques have been implemented for speaker identification and these techniques are reviewed in this section.

6.1. Deep neural network

Deep Neural Network (DNN) is a multi-layer feedforward neural network (MLFFNN) with several hidden layers (at least 3) (Tesauro, 1992) as shown in Fig. 8. The additional hidden layers of DNN allow features composition from low layers, therefore modelling complex data with less units as compared to Artificial Neural Networks. The DNN is trained layer-wise where every layer learns from preceding layer using gradient descent, i.e. output of single layer is used as an input of next layer. This process is repeated for all layers. Usually, this layer by layer training is unsupervised followed by inclusive supervised training. The DNNs weights are updated using stochastic gradient descent (SGD) as defined below

$$\Delta\omega_{ij}(t+1) = \Delta\omega_{ij}(t) + \eta\frac{\partial C}{\partial\omega_{ij}} \tag{6.1}$$

where η is the learning rate, ω_{ij} is the weight and C is the cost function related to the weights.

For large training data, DNN can be trained in several batches of small size without losing efficiency (LeCun, Bottou, Bengio, & Haffner, 1998). Nonetheless, training DNN with multiple layers and several hidden units is very difficult as the total parameters to be optimized are very large.

Few studies have utilized deep neural networks for speaker identification. In their recent study, Dhakal et al. (2019) implemented a DNN with one input layer of 56 neurons, three hidden layers of 50, 100 and 25 neurons respectively and one output layer of 2 neurons. The hidden units were transformed as Rectified Linear Units i.e. input was mapped to its corresponding activation value and output layer was designed as a logistical function for obtaining an output value from 0 to 1. In the training phase, cross-entropy was used as a loss function for

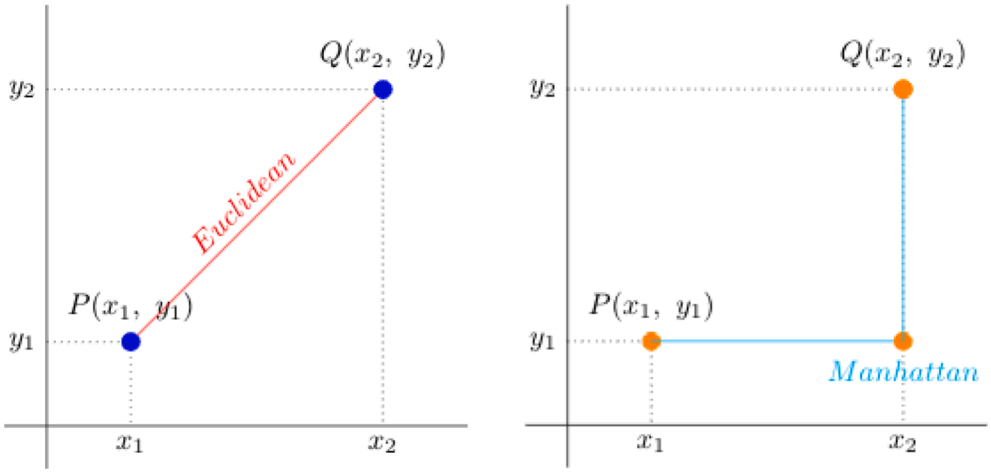

Fig. 8. Graphical representation of Euclidean and Manhattan distances.

backpropagating gradients which integrates the target probability, regulation parameter (L2) which penalizes the composite models and non-negative parameter to control the penalty magnitude. The training process started with random weights and then DNN reduced the loss function by continuously adjusting these weights. Moreover, the stochastic gradient descent (SGD) method was used for training the DNN. Finally, hybrid features (GF + CNN + Statistical) were fed to DNN for classification instead of extracting features. In addition, Zhang et al. (2015) implemented 5-layered DNN to extract bottleneck features instead of using it as classifier.

6.2. Convolution neural network

Convolutional Neural Network (CNN) is one of the most commonly used DL technique for feature representation as it provides salient, automatic and translational invariant set of features for various application domains (LeCun, Bengio, & Hinton, 2015). A CNN model consists of convolutional, pooling and fully connected (FC) layers joined together for extracting locally correlated features from speech data. The convolutional layers use different kernel sizes and strides to capture the features map and next pooled to reduce the number of connections between convolutional and pooling layer. Similarly, pooling layer minimize the features map, number of parameters enables the neural network to be translational invariant to distortion and changes. Various pooling techniques have been recommended for CNN implementation such as max-pooling, stochastic pooling, average pooling and spatial pooling (Guo et al., 2016). The FC layer is coupled with inference engine like SVM, HMM or Multinomial regression (SoftMax) that takes the feature vectors for identification of speaker. Moreover, the training of CNN involves the fine-tuning of hyper-parameters such as momentum, learning rate, weight decay, preliminary values of weight and mechanism to update weight (Hinton, 2012). In addition, activation functions, optimization algorithms, number of training epochs, mini-batch size and number of layers are also important in the training of CNN.

As CNN is most widely used DL technique, therefore several other well-known CNN models have been implemented such as AlexNet (Krizhevsky, Sutskever, & Hinton, 2012), Inception (Szegedy et al., 2015), ResNet (He, Zhang, Ren, & Sun, 2016), VGG (Simonyan & Zisserman, 2014) and DCGAN for image recognition and speech classification. The CNN architecture is shown in Fig. 9.

As recent studies have reported, automatically extracted features can significantly outperform ML classifiers trained on conventional handcrafted features and are more generalizable and robust when countering problems that include inbuilt noise. Therefore, Lukic et al. (2016) computed mel-spectrogram for all sentences of input data. With a sample rate of 16 kHz, FFT window length of 1024 samples and hop length of 160 samples, authors performed dynamic range compression of mel-spectrograms by using the elementwise function (Dieleman & Schrauwen, 2014). Next, a second-long snippets of non-overlapping fragments were extracted from the spectrograms and these 128 × 100 pixels images were used as input to the CNN. The network was comprised of two convolutional layers with 32 and 64 filters respectively. Each convolutional layer was followed by a max-pool layer with size 4 × 4 and stride of 2 × 2. Finally, two dense layers were used on the top of convolutional and pooling layers. In addition, rectified linear unit (ReLU) was used as activation function in each layer and softMax to compute the output. To prevent overfitting, a dropout layer was included between the dense layers with a rate of 0.5.

In another study, Imran et al. (2019) investigated the best and most efficient representation of acoustic feature maps for two-dimensional CNN by comparing different acoustic features on the RSR2015, MOOC and spoken numbers datasets. The authors utilized MFSC, MFCC together with delta and delta-delta coefficients and spectrograms from raw speech were utilized. Moreover, three different representations of spectrogram including padding, resize, and segmentation were fed as an input to the neural network. The resized spectrograms were found better to preserve the acoustic data with an accuracy of 98% on spoken numbers dataset in comparison to MFCC. The lowest accuracy was obtained for MFSC. Similar accuracy was obtained on MOOC and RSR2015 datasets with raw spectrograms.

In a recent study, Dhakal et al. (2019) proposed a novel architecture to enhance the performance of SI by utilizing the advantage of hybrid feature extraction methods that include Gabor Filter (GF) features, features extracted through CNN and statistical features as one matrix. The CNN was comprised of two convolutional layers with filter size 5 × 5, two pooling layers with filter size 2 × 2 and one fully-connected layer. The grayscale images of size 28 × 28 were fed as an input to the CNN. The hybrid feature set was classified using three classifiers namely DNN, random forest (RF) and SVM. The reported experimental results revealed that RF outperformed other two classifiers by achieving an accuracy of 94.87% on ELSDSR dataset.

In study by An et al. (2019a), instead of implementing CNN model from scratch for learning feature representation from speech data, the authors formulated speech identification problem as an image classification task. This shift provided the benefit of utilizing two existing pretrained CNNs methods namely VGG (Simonyan & Zisserman, 2014) and ResNets (He et al., 2016), recognized for their great performance in the classification of large-scale images and speech recognition applications (Hershey et al., 2017), (Deng, Eyben, Schuller, & Burkhardt, 2017). The structured self-attention layer (Lin et al., 2017) was used on the top of ResNets and VGG-like CNN in order to encode the variable length of speech sequence into a embedding matrix of fixed-size, followed by a temporal pooling layer. The authors investigate the effect of three types of pooling layers, i.e., max-pooling, average pooling and the combination of standard deviation pooling and average pooling on the proposed ResNet and VGG. Nonetheless, the average pooling layer achieved best performance on the proposed networks. Finally, softMax layer was used for classification of speech utterances. Moreover, the authors also

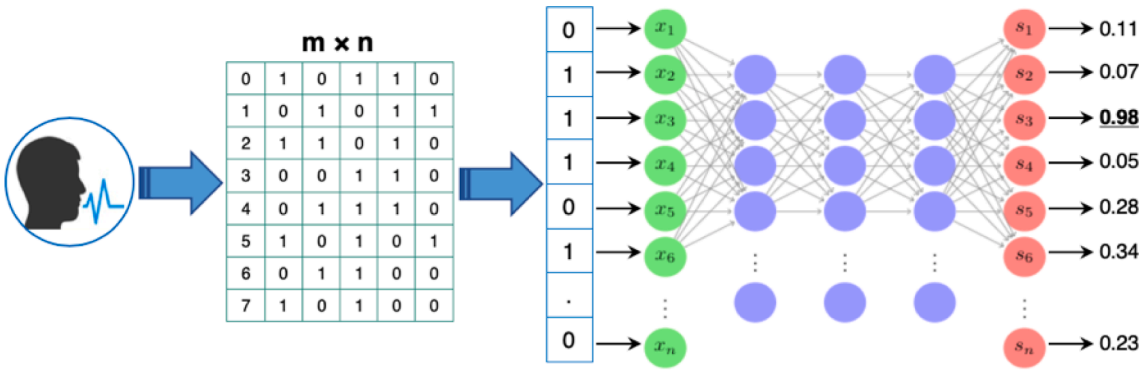

Fig. 9. Deep Neural Network Architecture for SI (Jahangir et al., 2020).

examine the effect of FBank acoustic features, spectrogram and MFCCs features on the proposed networks and found that filterbank features outperformed both MFCCs features and spectrogram by using large-scale VoxCeleb dataset.

In order to reduce training time, Yadav and Rai (2018) proposed a CNN at the top of popular VGG with significant changes to accommodate spectrogram inputs of variable length. The network was trained under the combined supervision of Center loss and SoftMax loss to get highly discriminative features suitable for SI. The experimental results using VoxCeleb dataset demonstrated that proposed model outperformed the baseline machine learning methods with few numbers of parameters.

6.3. Recurrent neural network

Recurrent Neural network (RNN) was proposed to model and analyze sequential data and implement temporal layer for learning composite variation in time-series data, where inputs and outputs are interdependent. Usually, this interdependency is helpful in estimating the possible state of the input. RNN estimate the disparity in sequential information by means of hidden unit cell. Fig. 10 shows the RNN architecture where $x_{0,1,2,\cdots,t}$ is the input, $h_{0,1,2,\cdots,t}$ is the hidden state, $y_{0,1,2,\cdots,t}$ is the output at time t and u, v, w are parameters for hidden units and their values are constantly updated for each time t to reveal the current position of network. The hidden state is computed as $h_t = f\left(u_{(x_t)} + w_{(h_{t-1})}\right)$ which means that next hidden state is estimated as activation of previous hidden state.Fig. 11.

RNN is suitable deep learning technique for speaker identification due to short-time level framing of signals for features extraction. However, training RNN is challenging due to vanishing gradients that affects its overall performance (Weninger, Ringeval, Marchi, & Schuller, 2016). To overcome this vanishing gradient problem, Recently, Hochreiter and Schmidhuber (1997) proposed long short term memory (LSTM) for modelling long-term temporal dependencies and integrate a cell among recurrent connections. All cells of memory capture the model temporal sequence, and integrate diversities of gates like input, output and function gate along learnable weights for controlling the inflow of new data. The residual connections are typically very deep and thus suitable for reducing the gradient problem.

Recently, several studies have employed RNN for feature representation in automatic speaker identification. For instance, Larsson (2014) extracted MFCC features along with ΔMFCC and ΔΔMFCC from audio files and fed as input to LSTM for speaker identification. These features were extracted from audio files by using a 25 ms window to form a frame. Then, the window was moved 10 ms at a time till the end of speech signal to reduce the risk of information lose between two consecutive frames. From each frame, 13 MFCC features were extracted together with ΔMFCC and ΔΔMFCC to better model a signal's behavior. Finally, a feature vector of size 39 related to single audio file were fed to input layer of size 39. Thus, LSTM was trained in such a way that each sequence of voice contained one target speaker. So, sequence-wise classification was performed instead of frame-wise classification. In addition, multiple networks were trained having different number of hidden layers and memory blocks. The best performance was achieved having 2 hidden layers and each layer with 5 memory blocks.

The above discussed study used human-driven methods for features extraction from each speech signal. However, in a recent study by Jung et al. (2018) proposed the use of raw waveforms were used as input for investigating the issue of overfitting in speaker identification model. The proposed model utilized convolution and pooling layers to withdraw a feature map from raw waveform. Then, the LSTM layer with 512 memory cells, was exploited to perform sequential modelling and to integrate the speaker feature followed by 2 fully connected layers with 1024 neurons and an output layer. The proposed model achieved 98.3% accuracy by using VoxCeleb dataset.

6.4. Restricted Boltzmann Machine

Restricted Boltzmann Machine (RBM) (Fischer & Igel, 2014; Hinton & Sejnowski, 1986) is a generative DL technique that trains through contrastive divergence to deliver unbiased estimation of maximum probability learning. Conversely, during training RMB methods are difficult to converge to local minimal, various data representation and hyper-parameters settings have been proposed in literatures to get best performance improvements (Cho, Raiko, & Ihler, 2011). To solve this issue, various techniques such as temperature-based RBM (Al-Rfou et al., 2016) and regularization by using noisy ReLU (Nair & Hinton, 2010) have been proposed. Two well-known RBM approaches are Deep Boltzmann Machine and Deep Belief Network (DBN). Deep Belief Networks proposed by Hinton, Osindero, and Teh (2006), is implemented by fusion of several RBMs that allows greedy layer-wise feature representation learning. It has several inter-connected hidden layers where output of each layer is used as an input to the next layer and is visible only to the next layer. Firstly, the initial layer gets the input data and the output after using activation function is feed as an input to the second layer and the process continues. Therefore, each layer in DBN is considered as an independent neural network with single hidden layer. The input data transformation can be achieved by using sampling or activation function. The next hidden layer thus becomes a visible layer for current hidden layer in order to train it as an RBM.

RBM is widely applied in speaker identification systems. For instance, Ghahabi and Hernando (2018) employed Universal RBM (URBM) to discover the total speaker and session variability among the background GMM supervectors assumed as visible units. GMM supervectors were created from warped spectral feature vectors given the UBM. Then, URBM was trained with variant of ReLU called variable ReLU (VReLU) for the transformation of inputs. In VReLU activation function, the unit values below threshold value are zeroed out instead of the permanent threshold 0 in ReLU. The weight matrix of visible–hidden connection was utilized to transform unknown GMM supervectors into low dimensional vectors after training URBM. Although, traditional i-vectors performed better than proposed GMM-RBM vectors with both PLDA and cosine scoring but the fusion of traditional i-vectors and proposed GMM-RBM vectors outperformed traditional i-vectors using NIST 2010 SRE dataset with less computational load.

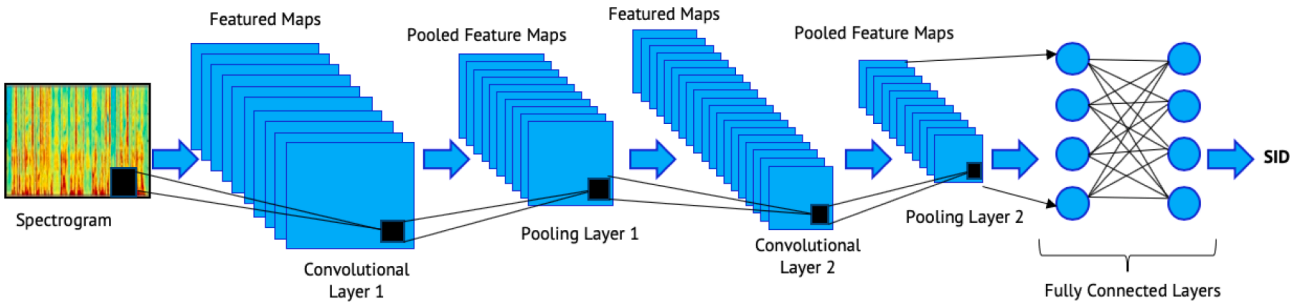

Fig. 10. . Convolution Neural Network Architecture.

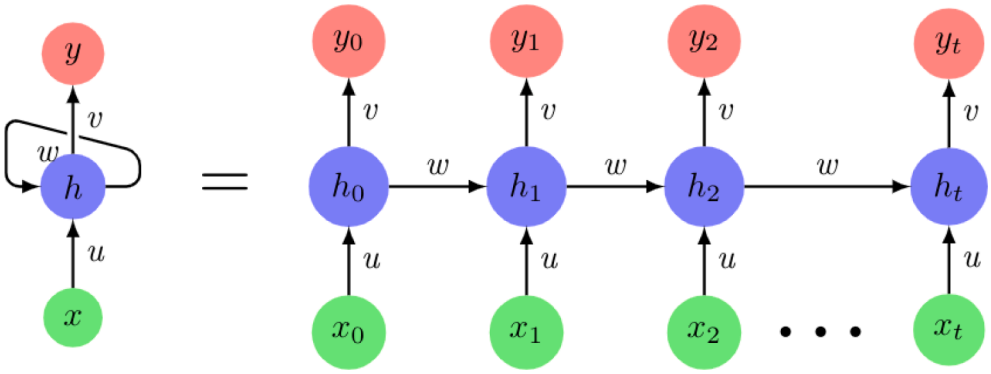

Fig. 11. . Recurrent Neural Network Architecture.

Table 12
. Deep Learning Techniques for SI.

Technique	Strengths	Weaknesses	Reference
DBN	Effectively yield discriminative features that estimate the complex non-linear dependency among features in each speech sample.	slow learning process and labeled dataset is required for training	Ali et al. (2018), Sun et al. (2019)
RBM	Automatically identify relationships between features, describing them in its weights which can be translated as basis vectors.	Not suitable for massive databases because of training and computational complexity.	Ghahabi and Hernando (2018)
DAE	Useful for unbalanced speech database and able to learn effective non-linear features.	Overfitting can occur because network parameters are more than input data.	Novotný et al. (2019), Tirumala and Shahamiri (2017), Zhang et al. (2015)
CNN	robust to noise, does not need pre-processing steps and capable to learn local correlation patterns in input features	Require massive amount of data and it is not efficient when modelling temporal dependencies in the speech signals.	Abdul (2019), An et al. (2019a), Bunrit et al. (2019), Hajavi and Etemad (2019), Imran et al. (2019), Liu et al. (2018), Lukic et al. (2016), Yadav and Rai (2018), Zhang et al. (2018a)
LSTM	deal with variable length inputs and effectively deal with vanishing gradient problem	Long processing time and computational cost because of updating a number of parameters.	Dovydaitis and Rudžionis (2018), Jung et al. (2018), Larsson (2014)
DNN	Excellent ability to deeply manipulate high-dimensional data.	Sensitive to the variation of input data and training process is slow	Dhakal et al. (2019), Zhang et al. (2015)

Representation learning from voice data has shown benefits over the human-driven features such as MFCCs in several acoustic applications. In many of the representation learning techniques, the connectionist models have been employed to learn latent features from fixed size data. For voice scripts of variable lengths, Ali et al. (2018) proposed a method to combine the MFCC features and learned features for speaker identification task. The dimensions of spectrograms generated from Urdu dataset were reduced by applying PCA technique to transform the frequencies into a low-dimensional space. Each PCA-transformed frequency was used as input to DBN to learn the latent features from voice data. Finally, the learned and MFCC features were fed to SVM classifier and 92.6% accuracy was achieved.

Since extracting features from GMM or DNN has made great progress in speaker identification, Sun et al. (2019) proposed a novel DGCS feature on the basis of a hybrid DBN–GMM model. In the proposed method, MFCC features extracted from speech signals were used as an input to DBN for gaining bottleneck features. The DBN was comprised of one input, two hidden, one output and one bottleneck layer in the middle of hidden layers with fewer nodes to withdraw discriminative information from MFCC features. Next, bottleneck features were used to train GMM for the extraction of deep Gaussian supervector (DGS), which was further transformed into DGCS by the supervector recombination method. Finally, SVM was trained by using DGCS to achieve classification.

6.5. Deep autoencoder

The classification task may not be effective when there is a large set of features included in a feature vector. The small set of features that best represents speakers is considered effective for the classification task. Thus, autoencoder are used to transform high-dimensional speech data into low-dimensional data. A deep autoencoder (DAE) consists of several autoencoder that are stacked together as layers. DAE is classified into two main components: encoder and decoder. Encoder converts the input into the hidden features whereas the decoder component reconstructs hidden features into almost exact representation to minimize the possibility of error rate. Both components are initially allocated with random weights and then trained by evaluating the difference between input data and output gained from encoding and decoding processes.

The in-depth observation of reviewed studies indicated that DAE is most widely used generative model because its ability to learn effective non-linear features. For instance, Tirumala and Shahamiri (2017) constructed a DAE network by stacking several autoencoder algorithms with distinct parametric values, led by a SoftMax layer for classification. The proposed DAE used three hidden layers with 16, 20 and 20 neurons in each layer respectively. The 16 MFCCs extracted from each audio file were fed as input to first layer. The proposed network outperformed baseline ANN by achieving 98.8% average accuracy on Census dataset.

7. Classification evaluation

There is a big variation in the performance measures employed for classification systems. Performance measures employed for critical study must be suitable for the classification system domain (Leonard, 2017). A confusion matrix (Table 12), can be employed to evaluate the performance of a classification problem on the base of test data. It is utilized for the prediction of negative and positive instances. TP (true positive) shows the instances in which actual and predicted classes are correct (i.e. positive) while TN (true negative) indicates where predicted and actual classes are negative. FN (false negative) represents the

instances, where the actual class is positive and predicted class is negative. Finally, FP shows the cases, where the actual class is negative and predicted class is positive. Optimal performance of SI system can be achieved by reducing FP and FN.

Several performance measures have been utilized for the classifier performance evaluation. The most widely used measures for automatic speaker identification are accuracy, recall, precision and F-measure. These performance measures are briefly presented below.

7.1. Accuracy

Accuracy is the most widely used performance measure the number of correctly classified instances by particular classification algorithm. It provides the ratio of correctly predicted instances against the entire number of instances. Eq. (7.1) shows the mathematical representation of accuracy.

$$Accuracy = \frac{(TP + TN)}{(TN + TP + FN + FP)} \tag{7.1}$$

If the classification algorithm accurately predicts half of the test instances, it is said to be 50% accurate. Commonly, it is assumed that we have higher classification power as the classifier accuracy increases. For instance, we have a classifier to detect the spam in emails with these values TN = 90, FN = 10, FP = 0 and TP = 0. From Eq. (7.1), accuracy $= (0 + 90)/(90 + 0 + 10 + 0) = 90\%$.. Although this classifier has high accuracy, but has zero classification power because it predicts only negative instances. This scenario reflects the Accuracy Paradox, which means that classifiers with a specified level of accuracy may have better classification power than classifiers with high accuracy. In some scenarios, accuracy of the classifier can be misleading to overfitting or underfitting if the dataset has an unequal number of utterances in each class. Therefore, to address this issue, this measure is best utilized together with the other performance measures such as precision and recall.

7.2. Precision

It is the ratio of the negative instances that are predicted as negative. Generally, the exactness is computed through the precision measure. The higher the values of precision, the lower the FP rate.

$$Precision = \frac{TP}{FP + TP} \tag{7.2}$$

7.3. Recall

Recall represents the ratio of number of true positives (TPs) or correct positive predictions against the total number of positives. Recall is also known as true positive rate (TPR).

$$Recall = \frac{TP}{FN + TP} \tag{7.3}$$

7.4. F-measure

It is the weighted harmonic mean of recall and precision mainly when there is extreme balance of FP and FN. The standard F-measure is F1, which gives equal important to recall and precision.

$$F - measure = 2\frac{Precision \times recall}{Precision + recall} \tag{7.4}$$

7.5. Equal error rate

ERR is used to find the common value for its false acceptance rate (FAR) and its false rejection rate (FRR). The lower EER value indicates the higher accuracy of the system. FAR and FRR can be calculated using Eq 7.5 and 7.6 and (Marcel, Nixon, & Li, 2014) while equal error rate (EER) can be calculated using Eq 7.7.

$$FAR = FPR = \frac{FP}{FP + TN} \tag{7.5}$$

$$FRR = FNR = \frac{FN}{FN + TP} \tag{7.6}$$

$$EER = \frac{FAR + FRR}{2} \tag{7.7}$$

7.6. Receiver operating characteristics

ROC curve represents the false acceptance rate (FAR) as a function of false rejection rate (FRR) for different values. The ROC curve is a visual plot that demonstrate the performance of classification algorithm with respect to the discrimination threshold. FAR and FRR vary with the size of training database and decision threshold for calculating scores. Thus, ROC curve plots false acceptance rate against the corresponding false rejection rate as a function of decision threshold. In addition, the results of classifier can be changed by changing this decision threshold (Almaadeed et al., 2015).

7.7. Root mean square error

RMSE is a standard parameter to calculate the error of a trained model in prediction of quantitative data. Formally it is written as follows

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{\left(\hat{y}_i - y_i\right)^2}{n}} \tag{7.8}$$

where $\hat{y}_1, \hat{y}_2, \hat{y}_3, \cdots, \hat{y}_n$ are the predicted values and $y_1, y_2, y_3, \cdots, y_n$ are the observed values.

The RMSE is an appropriate indicator to evaluate model performance when the distribution of error is likely to be Gaussian. However, it is not a good parameter to calculate the average performance of model (Chai & Draxler, 2014).

The reviewed articles revealed that the most widely used performance evaluation metric for speaker identification was accuracy followed by the EER as shown in Table 13. The potential reason for the extensive use of weighted accuracy by researchers may be because it provides the average accuracy of each class by computing the sum of accuracy per class to be predicted divided by the total number of classes.

$$AverageAccuracy = \frac{1}{N}\sum_{i=1}^{N} \frac{TP_i + TN_i}{TP_i + TN_i + FP_i + FN_i} \tag{7.9}$$

where N represents the number of classes. However, the accuracy performance measure can mislead to overfitting or underfitting for multiclass database and imbalance number of utterances.

8. Deep learning implementation frameworks

In recent years, deep learning techniques have gained popularity in a variety of applications. Thus, there has been a great deal of interest from a range of researchers and business groups to build software frameworks

Table 13
Confusion matrix.

	Actual instances	
	Yes	No
Predicted Instances		
Yes	TP	FN
No	FP	TN

Table 14
The frequency of evaluation measures in reviewed studies.

References	Accuracy	Precision	Recall	F1 Score	ERR	ROC	RMSE
Abdul (2019), Ali et al. (2018), An et al. (2019a), Bisio et al. (2018), Daqrouq (2011), Fan and Hansen (2010), Faragallah (2018), Jawarkar et al. (2015), Krobba et al. (2019), Medikonda and Madasu (2018), Mokgonyane et al. (2019), Renisha and Jayasree (2019), Sardar and Shirbahadurkar (2018a), Sardar and Shirbahadurkar (2018b), Sardar and Shirbahadurkar (2019), Soleymanpour and Marvi (2017), Sun et al. (2019), Wang et al. (2015), Wu and Tsai (2011), Zhang et al. (2015), Zhao et al. (2014)	✓	×	×	×	×	×	×
Imran et al. (2019), Jahangir et al. (2018), Manikandan and Chandra (2016)	✓	✓	✓	✓	×	×	×
Ghahabi and Hernando (2018), Hajavi and Etemad (2019), Jung et al. (2018), Liu et al. (2018), Lukic et al. (2016), Novotný et al. (2019), Yadav and Rai (2018), Zhang et al. (2018a), Zhang et al. (2018)	×	×	×	×	✓	×	×
Daqrouq and Tutunji (2015)	✓	×	×	×	✓	×	×
Indumathi and Chandra (2015)	✓	✓	✓	×	×	×	✓
Almaadeed et al. (2015)	✓	×	×	×	×	✓	×
Ajmera et al. (2011), Tirumala and Shahamiri (2017)	✓	×	×	×	×	×	✓
Abdalmalak and Gallardo-Antolín (2018)	×	×	×	×	×	✓	×

that will help to easily implement and evaluate different deep learning architectures. Some of the most widely used frameworks for deep learning are: Caffe, Torch, Theano, Neon, Chainer, TensorFlow, DeepLearning4J, DeepLearnToolbox, Matlab Deep Learning, MatConvNet etc. (for a complete list visit http://deeplearning.net/software links/). Most of these software frameworks are already mature as of today and are very effective in the training of deep neural networks by using GPUs to accelerate training process. Table 14 represents the DL implementation software frameworks that have demonstrated state-of-the-art performance and are useful for applications such as speaker identification. A number of software frameworks support a wide range of deep learning techniques such as CNNs, AEs, fully-connected networks (FCNs), RBMs and RNNs and implement popular activation functions and optimizers. Varying functionality has raised the issue of selecting an appropriate framework. For instance, DeepLearnToolbox (Palm, 2014), Theano (Al-Rfou et al., 2016), Caffe (Jia et al., 2014), MatConvNet (Vedaldi & Lenc, 2015), DeepLearning4J (Team, 2016) and CNTK (Seide & Agarwal, 2016) all support CNN but these frameworks vary in computational complexity, environment, efficiency, programming languages and hyper-parameters. It is noteworthy that DeepLearnToolbox, ConvNet (Demyanov), MatConvNet and MDLTB are solely implemented in MATLAB® which can lead to slow computation and performance issues when used for big data. Moreover, studies in speaker identification utilized these framework in their implementation and include Torch (Bahrampour, Ramakrishnan, Schott, & Shah, 2016; Kovalev, Kalinovsky, & Kovalev, 2016), Chainer (Tokui, Oono, Hido, & Clayton, 2015), and NeuralNetworks (Vasilev, 2019) represent a group of DL frameworks designed to provide high-performance CNN training using GPUs. The rest of frameworks implement a range of deep neural networks by means of other libraries. For example, ConvNet and Keras (Gulli & Pal, 2017) and are based on Tensorflow and Theano respectively. Table 14 represents each deep learning framework, GPU support capabilities, Platform support, deep learning techniques supports, optimization methods and activation functions.Table 15.

9. Future research challenges

This review article has identified several open research challenges inherent in the earlier studies in speaker identification area. The highlighted research issue requires significant research efforts to enhance the performance of speaker identification systems. These open research challenges are discussed below:

Creation of high-quality multilingual speech databases: The evaluation of AI techniques needs massive speech databases. The careful observation of the current review article shows that most of the reported studies used single language databases consisting of low number of speakers and utterances. Therefore, significant research effort can be exerted to construct massive multilingual speech databases through standard design criteria for performance enhancement.

Clustering-based Technique is mainly employed in pattern recognition applications but still an infant in the SI domain. Several researchers in the reviewed studies applied a supervised-learning technique to construct a classification model and achieved a good accuracy despite the limitations of such techniques. One of the most important limitations in supervised-learning is the database labelling in order to create the training sets. These tasks require a significant amount of time in the creation of database. Therefore, more research effort can be exerted on the clustering technique for modelling SI.

Transfer learning for SI using DL techniques: Transfer learning (TL) is a popular DL approach where pre-trained deep neural networks are employed as the initial point on new databases. This process reduces the training time and computing resources required to train a deep neural network for related problems. The most widely used TL deep networks include GoogleNet, AlexNet, Petri Net and LeNet. Although TL is fast, the performance may not be improved for related problems since the deep networks are pre-trained on various databases. Hence, these TL deep networks may provide low classification result in SI databases. Thus, an extensive amount of research can be conducted to design and implement a new TL architecture for speech databases.

Training with limited dataset: Large number speech utterances are generally required during training of DL model. It may not provide satisfactory classification results if the training data is small. This problem can be solved by using different augmentation methods, including flipping, rotation and cropping along with spectrograms. Data augmentation methods take advantage of small amount of speech data by transforming existing training data to create new data. Further researcher is required to generate ampler training spectrograms, so that, DL model could be trained efficiently and learn more discriminative features.

Real-time implementation of DL techniques for SI: Implementation of DL techniques in real-time will help to decrease the computational complexity on data transfer and storage. Nonetheless, this technique is controlled by memory constrain and data acquisition in current mobile devices. Moreover, the initialization and tuning of a high number of parameters in DL model boosts computational complexity and is not useful for low-energy mobile devices. Thus, utilizing methods like optimal compression and the use of smartphone enabled GPU to reduce computational time and consumptions of resources is highly required. The use of mobile cloud computing services to minimize training time and memory usage is another is another method that support real-time implementation of DL techniques for SI.

10. Conclusion

We have identified and presented detail of areas that build an automatic speaker identification system. The training of such systems requires utterances provided by speech databases which are recorded in different languages using different recording devices under quiet or

Table 15
Deep learning software frameworks.

Software	GPU	Lic. / Access	PLATFORM			Support	Supporting DL Techniques					Optimizer	Activation Function
			❖	LINUX			FCNN	AE	CNN	RNN	RBM		
Theano	✓	NumPy[a]	✓	✓	✓	Python	✓	DAE	✓	×	DBN	SGD	ReLU, Tanh SoftMax
Caffe	✓	BSD[b]	✓	✓	✓	Python MATLAB C++	×	×	✓	×	×	SGD	ReLU, Tanh Sigm, ELU
DeepLearn-Toolbox	✓	BSD[c]	✓	✓	×	MATLAB	✓	SAE, DAE	✓	×	DBN	BP	Sigmoid
MatConvNet	✓	BSD[d]	✓	✓	✓	MATLAB C++	×	×	✓	×	×	SGD	ReLU, Sigmoid
Tensorflow	✓	Apache[e]	✓	✓	✓	Python, C++	×	CAE	✓	✓	×	SGD BP	ReLU, Sigmoid
Keras	✓	MIT[f]	✓	✓	✓	Python	×	×	✓	✓	×	SGD ADAM RMSPrp	ELU, ReLU SoftMax SeLU, Tanh Sigmoid
CNTK	✓	MIT[g]	✓	✓	×	C++, C#, Python, Java	×	×	✓	✓	×	SGD	ReLU, Tanh ELU
PyBrain	✓	BSD[h]	✓	✓	✓	Python	✓	×	×	✓	✓	BP	Sigmoid
Torch	✓	BSD[i]	✓	✓	✓	Lua C	✓	✓	✓		×	SGD ADAM	ReLU, Tanh SoftMax
MDLTB	✓		✓	✓	✓	MATLAB	✓	SAE, DAE	✓	✓	×	ADAM	ReLU, PReLU
DL4J	✓	Apache[j]	✓	✓	✓	Scala Java	✓	✓	✓	✓	×	SGD ADAM RMSPrp	Tanh, ReLU SoftM, ELU Sigmoid
Chainer	✓	MIT[k]	✓	✓	✓	Python	✓	✓	✓	✓	×	SGD	ReLU
Neural Networks	✓	MIT[l]	✓	✓	✓	Java	✓	SAE, DAE	✓	×	RBM DBN	-	ReLU, Tanh SoftM, LRN Sigm
ConvNet	✓	BSD[m]	✓	✓	×	MATLAB	×	×	✓	×	×	IBP	ReLU, Sigm SoftMax
OpenNN	✓	GNU[n]	✓	✓	✓	Python C++	✓	×	×	×	×	-	-
RNNLIB	✓	BSD[o]	×	✓	✓	C++	×	×	×	✓	×	-	-
SimpleDNN	✓	mozilla[p]	✓	✓	✓	Kotlin	✓	×	×	✓	×	-	ELU, SoftMax

** MDLTB = MATLAB Deep Learning Toolbox, DL4J = DeepLearning4J, SoftM = SoftMax, Sigm = Sigmoid.
[a] http://deeplearning.net/software/theano/.
[b] https://github.com/BVLC/caffe/.
[c] https://github.com/rasmusbergpalm/DeepLearnToolbox.
[d] http://www.vlfeat.org/matconvnet/.
[e] www.tensorflow.org.
[f] https://github.com/keras-team/keras.
[g] https://github.com/Microsoft/cntk.
[h] https://github.com/pybrain/pybrain.
[i] https://github.com/torch/torch7.
[j] https://github.com/eclipse/deeplearning4j.
[k] https://github.com/chainer/chainer.
[l] https://github.com/ivan-vasilev/neuralnetworks.
[m] https://github.com/sdemyanov/ConvNet.
[n] http://www.opennn.net/download/index.html.
[o] https://github.com/cudamat/cudamat.
[p] https://github.com/KotlinNLP/SimpleDNN.

noisy environment. The speech signals are then pre-processed to make them suitable for feature extraction and the development of an efficient and robust SI system. SI systems most widely use spectral features, statistical features, short-term and voice source, and linear prediction-based features since they yield better classification results which can further be enhanced by combining various features. Several open source toolkits or libraries are available which can extract various acoustic features from speech signals. We provided a number of feature extraction toolkits that can be used instead of writing complex codes from scratch. After extraction of all features, a classification algorithm is selected from wide variety for SI systems. Although most use conventional ML approaches but a good number of studies employed recent DL advances such as CNN, RNN, DAE, DBN and RBM etc.

On the other hand, evaluation matrices are essential to ensure generalization within speech databases. These metrices are implemented to avoid model overfitting on the training data. Similarly, to implement DL techniques for SI, various software frameworks and hardware technologies such as GPUs have been developed. Some of these frameworks have been released to the research community as open source projects. We provide these software frameworks considering their characteristics and what determines the choice of developers to use such frameworks. Finally, we presented open research challenges that require significant research efforts and improvements in the field of SI systems.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This work was supported in part by the Ministry of Higher Education Malaysia under Grant FRGS-FP111-2018A.

References

Abdalmalak, K. A., & Gallardo-Antolín, A. (2018). Enhancement of a text-independent speaker verification system by using feature combination and parallel structure classifiers. *Neural Computing and Applications, 29*(3), 637–651.

Abdul, Z. K. (2019). Kurdish speaker identification based on one dimensional convolutional neural network. *Computational Methods for Differential Equations, 7*, 566–572.

Ahmad, K. S., Thosar, A. S., Nirmal, J. H., & Pande, V. S. (2015). A unique approach in text independent speaker recognition using MFCC feature sets and probabilistic neural network. In 2015 Eighth International Conference on Advances in Pattern Recognition (ICAPR) (pp. 1–6): IEEE.

Ajmera, P. K., Jadhav, D. V., & Holambe, R. S. (2011). Text-independent speaker identification using Radon and discrete cosine transforms based features from speech spectrogram. *Pattern Recognition, 44*(10-11), 2749–2759.

Al-Rawahy, S., Hossen, A., & Heute, U. (2012a). Text-independent speaker identification system based on the histogram of DCT-cepstrum coefficients. *International Journal of Knowledge-based and Intelligent Engineering Systems, 16*(3), 141–161.

Al-Rawahy, S., Hossen, A., & Heute, U. (2012b). Text-independent speaker identification system based on the histogram of DCT-cepstrum coefficients. *International Journal of Knowledge-based and Intelligent Engineering Systems, 16*(3), 141–161.

Al-Rfou, R., Alain, G., Almahairi, A., Angermueller, C., Bahdanau, D., Ballas, N., Bastien, F., Bayer, J., Belikov, A., & Belopolsky, A. (2016). Theano: A Python framework for fast computation of mathematical expressions. arXiv preprint arXiv:1605.02688.

Ali, H., Tran, S. N., Benetos, E., & Garcez, A. S. (2018). Speaker recognition with hybrid features from a deep belief network. *Neural Computing and Applications, 29*, 13–19.

Almaadeed, N., Aggoun, A., & Amira, A. (2015). Speaker identification using multimodal neural networks and wavelet analysis. *IET Biometrics, 4*(1), 18–28.

Almaadeed, N., Aggoun, A., & Amira, A. (2016). Text-independent speaker identification using vowel formants. *Journal of Signal Processing Systems, 82*(3), 345–356.

Alsulaiman, M., Muhammad, G., Bencherif, M. A., Mahmood, A., & Ali, Z. (2013). KSU rich Arabic speech database. Information (Japan), 16, 4231–4253.

An, N. N., Thanh, N. Q., & Liu, Y. (2019a). Deep CNNs with Self-Attention for Speaker Identification. IEEE Access.

Arons, B. M. (1994). Interactively skimming recorded speech. Massachusetts Institute of Technology.

Avci, D. (2009). An expert system for speaker identification using adaptive wavelet sure entropy. *Expert Systems with Applications, 36*(3), 6295–6300.

Badshah, A. M., Rahim, N., Ullah, N., Ahmad, J., Muhammad, K., Lee, M. Y., Kwon, S., & Baik, S. W. (2019). Deep features-based speech emotion recognition for smart affective services. Multimedia Tools and Applications, 78, 5571–5589.

Bahrampour, S., Ramakrishnan, N., Schott, L., & Shah, M. (2016). Comparative study of caffe, neon, theano, and torch for deep learning.

Benesty, J., Sondhi, M. M., & Huang, Y. A. (2008). Introduction to speech processing. In Springer Handbook of Speech Processing (pp. 1–4): Springer.

Bernardo, J., Bayarri, M., Berger, J., Dawid, A., Heckerman, D., Smith, A., & West, M. (2007). Generative or discriminative? getting the best of both worlds. *Bayesian Statistics, 8*, 3–24.

Bisio, I., Garibotto, C., Grattarola, A., Lavagetto, F., & Sciarrone, A. (2018). Smart and robust speaker recognition for context-aware in-vehicle applications. IEEE Transactions on Vehicular Technology, 67, 8808–8821.

Blazek, R. B., & Hong, W.-T. (2012). Robust Hierarchical Linear Model Comparison for End-of-Utterance Detection under Noisy Environments. In 2012 International Symposium on Biometrics and Security Technologies (pp. 126–133): IEEE.

Bogdanov, D., Wack, N., Gómez Gutiérrez, E., Gulati, S., Herrera Boyer, P., Mayor, O., Roma Trepat, G., Salamon, J., Zapata González, J. R., & Serra, X. (2013). Essentia: An audio analysis library for music information retrieval. In Britto A, Gouyon F, Dixon S, editors. 14th Conference of the International Society for Music Information Retrieval (ISMIR); 2013 Nov 4-8; Curitiba, Brazil.[place unknown]: ISMIR; 2013. p. 493-8.: International Society for Music Information Retrieval (ISMIR).

Bou-Ghazale, S. E., & Assaleh, K. (2002). A robust endpoint detection of speech for noisy environments with application to automatic speech recognition. In 2002 IEEE International Conference on Acoustics, Speech, and Signal Processing (Vol. 4, pp. IV-3808-IV-3811): IEEE.

Bullock, J., & Conservatoire, U. (2007). Libxtract: a Lightweight Library for audio Feature Extraction. In ICMC.

Bunrit, S., Inkian, T., Kerdprasop, N., & Kerdprasop, K. (2019). Text-independent speaker identification using deep learning model of convolution neural network. *International Journal of Machine Learning and Computing, 9*(2), 143–148.

Calzà, L., Gagliardi, G., Favretti, R. R., & Tamburini, F. (2020). Linguistic features and automatic classifiers for identifying mild cognitive impairment and dementia. *Computer Speech & Language, 65*, Article 101113.

Campbell, J., & Higgins, A. J. L. D. C., Philadelphia. (1994). YOHO speaker verification.

Campbell, J. P., Shen, W., Campbell, W. M., Schwartz, R., Bonastre, J.-F., & Matrouf, D. (2009). Forensic speaker recognition. *IEEE Signal Processing Magazine, 26*(2), 95–103.

Chai, T., & Draxler, R. R. (2014). Root mean square error (RMSE) or mean absolute error (MAE)? *Geoscientific Model Development Discussions, 7*(1), 1525–1534.

Chakroborty, S., & Saha, G. (2009). Improved text-independent speaker identification using fused MFCC & IMFCC feature sets based on Gaussian filter. *International Journal of Signal Processing, 5*, 11–19.

Cho, K., Raiko, T., & Ihler, A. T. (2011). Enhanced gradient and adaptive learning rate for training restricted Boltzmann machines. In Proceedings of the 28th international conference on machine learning (ICML-11) (pp. 105–112).

Cortes, C., & Vapnik, V. (1995). Support-vector networks. Machine learning, 20, 273–297.

Covoes, T. F., & Hruschka, E. R. (2013). Unsupervised learning of gaussian mixture models: Evolutionary create and eliminate for expectation maximization algorithm. In 2013 IEEE Congress on Evolutionary Computation (pp. 3206–3213): IEEE.

Cummins, F., Grimaldi, M., Leonard, T., & Simko, J. (2006). The chains speech corpus: Characterizing individual speakers. In Proc of SPECOM (pp. 1–6).

Daqrouq, K. (2011). Wavelet entropy and neural network for text-independent speaker identification. *Engineering Applications of Artificial Intelligence, 24*(5), 796–802.

Daqrouq, K., & Tutunji, T. A. (2015). Speaker identification using vowels features through a combined method of formants, wavelets, and neural network classifiers. *Applied Soft Computing, 27*, 231–239.

Deller, J. R., Proakis, J. G., & Hansen, J. H. (2000). Discrete-time processing of speech signals. In: Institute of Electrical and Electronics Engineers.

Demyanov, S. ConvNet. URL: http://github. com/sdemyanov/ConvNet (visited on 04/22/2015).

Deng, J., Eyben, F., Schuller, B., & Burkhardt, F. (2017). Deep neural networks for anger detection from real life speech data. In 2017 Seventh International Conference on Affective Computing and Intelligent Interaction Workshops and Demos (ACIIW) (pp. 1-6): IEEE.

Dhakal, P., Damacharla, P., Javaid, A. Y., & Devabhaktuni, V. (2019). A Near Real-Time Automatic Speaker Recognition Architecture for Voice-Based User Interface. Machine Learning and Knowledge Extraction, 1, 504–520.

Dieleman, S., & Schrauwen, B. (2014). End-to-end learning for music audio. In 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 6964–6968): IEEE.

Disken, G., Tufekci, Z., Saribulut, L., & Cevik, U. (2017). A review on feature extraction for speaker recognition under degraded conditions. IETE Technical Review, 34, 321–332.

Doddington, G. (2012). The effect of target/non-target age difference on speaker recognition performance. In Odyssey 2012-The Speaker and Language Recognition Workshop.

Doddington, G. R., Przybocki, M. A., Martin, A. F., & Reynolds, D. A. (2000). The NIST speaker recognition evaluation–overview, methodology, systems, results, perspective. Speech Communication, 31, 225–254.

Dovydaitis, L., & Rudžionis, V. E. (2018). Building LSTM neural network based speaker identification system. *Computational Science and Techniques*, 574–580.

Dutta, M., Patgiri, C., Sarma, M., & Sarma, K. K. (2015). Closed-set text-independent speaker identification system using multiple ann classifiers. In Proceedings of the 3rd

International Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA) 2014 (pp. 377–385): Springer.
Eyben, F., Weninger, F., Gross, F., & Schuller, B. (2013). Recent developments in opensmile, the munich open-source multimedia feature extractor. In Proceedings of the 21st ACM international conference on Multimedia (pp. 835-838): ACM.
Falcone, M., & Gallo, A. (1996). The“ siva” speech database for speaker verification: Description and evaluation. In Proceeding of Fourth International Conference on Spoken Language Processing. ICSLP'96 (Vol. 3, pp. 1902–1905): IEEE.
Fan, X., & Hansen, J. H. (2010). Speaker identification within whispered speech audio streams. *IEEE Transactions on Audio, Speech, and Language Processing, 19*, 1408–1421.
Faragallah, O. S. (2018). Robust noise MKMFCC–SVM automatic speaker identification. *International Journal of Speech Technology, 21*(2), 185–192.
Faundez-Zanuy, M., Hagmüller, M., & Kubin, G. (2007). Speaker identification security improvement by means of speech watermarking. Pattern Recognition, 40, 3027–3034.
Feng, L., & Hansen, L. K. (2005). A new database for speaker recognition: IMM, Informatik og Matematisk Modelling, DTU.
Fierrez, J., Morales, A., Vera-Rodriguez, R., & Camacho, D. (2018). Multiple classifiers in biometrics. Part 1: Fundamentals and review. *Information Fusion, 44*, 57–64.
Figo, D., Diniz, P. C., Ferreira, D. R., & Cardoso, João. M. P. (2010). Preprocessing techniques for context recognition from accelerometer data. *Personal and Ubiquitous Computing, 14*(7), 645–662.
Fischer, A., & Igel, C. (2014). Training restricted Boltzmann machines: An introduction. *Pattern Recognition, 47*(1), 25–39.
Friedl, M. A., & Brodley, C. E. (1997). Decision tree classification of land cover from remotely sensed data. *Remote Sensing of Environment, 61*(3), 399–409.
Georgescu, M.-I., Ionescu, R. T., & Popescu, M. (2019). Local learning with deep and handcrafted features for facial expression recognition. *IEEE Access, 7*, 64827–64836.
Ghahabi, O., & Hernando, J. (2018). Restricted Boltzmann machines for vector representation of speech in speaker recognition. *Computer Speech & Language, 47*, 16–29.
Giannakopoulos, T. (2015). pyaudioanalysis: An open-source python library for audio signal analysis. *PLoS ONE, 10*(12), e0144610.
Gill, M. K., Kaur, R., & Kaur, J. (2010). Vector quantization based speaker identification. *International Journal of Computer Applications, 4*(2), 1–4.
Godfrey, J. J., Holliman, E. C., & McDaniel, J. (1992). SWITCHBOARD: Telephone speech corpus for research and development. In [Proceedings] ICASSP-92: 1992 IEEE International Conference on Acoustics, Speech, and Signal Processing (Vol. 1, pp. 517–520): IEEE.
Gomar, M. G. (2015). System and method for speaker recognition on mobile devices. In: Google Patents.
Gulli, A., & Pal, S. (2017). Deep Learning with Keras: Packt Publishing Ltd.
Guo, Y., Liu, Y., Oerlemans, A., Lao, S., Wu, S., & Lew, M. S. (2016). Deep learning for visual understanding: A review. *Neurocomputing, 187*, 27–48.
Hajavi, A., & Etemad, A. (2019). A deep neural network for short-segment speaker recognition. arXiv preprint arXiv:1907.10420.
Hansen, J. H., Sarikaya, R., Yapanel, U., & Pellom, B. (2001). Robust speech recognition in noise: an evaluation using the SPINE corpus. In Seventh European Conference on Speech Communication and Technology.
He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770–778).
He, L., Lech, M., Maddage, N., & Allen, N. (2009). Emotion recognition in speech of parents of depressed adolescents. In 2009 3rd International Conference on Bioinformatics and Biomedical Engineering (pp. 1–4): IEEE.
He, L., Lech, M., Memon, S., & Allen, N. (2008). Recognition of stress in speech using wavelet analysis and teager energy operator. In Ninth Annual Conference of the International Speech Communication Association.
Hennebert, J., Melin, H., Petrovska, D., & Genoud, D. (2000). POLYCOST: A telephone-speech database for speaker recognition. *Speech Communication, 31*(2-3), 265–270.
Hennebert, J., Melin, H., Petrovska, D., & Genoud, D. J. S. c. (2000b). POLYCOST: a telephone-speech database for speaker recognition. 31, 265–270.
Hershey, S., Chaudhuri, S., Ellis, D. P., Gemmeke, J. F., Jansen, A., Moore, R. C., Plakal, M., Platt, D., Saurous, R. A., & Seybold, B. (2017). CNN architectures for large-scale audio classification. In 2017 ieee international conference on acoustics, speech and signal processing (icassp) (pp. 131–135): IEEE.
Hinton, G. E. (2012). A practical guide to training restricted Boltzmann machines. In Neural networks: Tricks of the trade (pp. 599–619): Springer.
Hinton, G. E., Osindero, S., & Teh, Y.-W. (2006). A fast learning algorithm for deep belief nets. *Neural Computation, 18*(7), 1527–1554.
Hinton, G. E., & Sejnowski, T. J. (1986). Learning and relearning in Boltzmann machines. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, 1*, 2.
Hochreiter, S., & Schmidhuber, Jürgen (1997). Long short-term memory. *Neural Computation, 9*(8), 1735–1780.
Huang, Y., Tian, K., Wu, A., & Zhang, G. (2019). Feature fusion methods research based on deep belief networks for speech emotion recognition under noise condition. *Journal of Ambient Intelligence and Humanized Computing, 10*(5), 1787–1798.
Hunt, A. K., & Schalk, T. B. (1996). Simultaneous voice recognition and verification to allow access to telephone network services. In: Google Patents.
Hwang, I., Park, H.-M., & Chang, J.-H. (2016). Ensemble of deep neural networks using acoustic environment classification for statistical model-based voice activity detection. *Computer Speech & Language, 38*, 1–12.
Imran, A. S., Haflan, V., Shahrebabaki, A. S., Olfati, N., & Svendsen, T. K. (2019). Evaluating Acoustic Feature Maps in 2D-CNN for Speaker Identification. In Proceedings of the 2019 11th International Conference on Machine Learning and Computing (pp. 211–216): ACM.
Indumathi, A., & Chandra, E. (2015). Speaker identification using bagging techniques. In 2015 International Conference on Computers, Communications, and Systems (ICCCS) (pp. 223–229): IEEE.
Islam, M., & Rahman, M. (2009). Improvement of text dependent speaker identification system using neuro-genetic hybrid algorithm in office environmental conditions. arXiv preprint arXiv:0909.2363.
Jagdale, S., Shinde, A., & Chitode, J. (2020). Robust Speaker Recognition Based on Low-Level-and Prosodic-Level-Features. In Advances in Data Sciences, Security and Applications (pp. 267–274): Springer.
Jahangir, R., Teh, Y. W., Ishtiaq, U., Mujtaba, G., & Nweke, H. F. (2018). Automatic Speaker Identification through Robust Time Domain Features and Hierarchical Classification Approach. In Proceedings of the International Conference on Data Processing and Applications (pp. 34–38): ACM.
Jahangir, R., TEh, Y. W., Memon, N. A., Mujtaba, G., Zareei, M., Ishtiaq, U., … Ali, I. (2020). Text-independent speaker identification through feature fusion and deep neural network. *IEEE Access, 8*, 32187–32202.
Jasmine, J., Sandhya, S., Ravichandran, K., & Balasubramaniam, D. (2016). Silence Removal from Audio Signal Using Framing and Windowing Method and Analyze Various Parameter. International Journal of Innovative Research In Computer And Communication Engineering, 4.
Jawarkar, N. P., Holambe, R. S., & Basu, T. K. (2015). Effect of nonlinear compression function on the performance of the speaker identification system under noisy conditions. In Proceedings of the 2nd International Conference on Perception and Machine Intelligence (pp. 137–144): ACM.
Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., & Darrell, T. (2014). Caffe: Convolutional architecture for fast feature embedding. In Proceedings of the 22nd ACM international conference on Multimedia (pp. 675–678): ACM.
Jiang, D.-N., Lu, L., Zhang, H.-J., Tao, J.-H., & Cai, L.-H. (2002). Music type classification by spectral contrast feature. In Proceedings. IEEE International Conference on Multimedia and Expo (Vol. 1, pp. 113–116): IEEE.
John S. Garofolo, Lori F. Lamel, William M. Fisher, Jonathan G. Fiscus, David S. Pallett, Nancy L. Dahlgren, & Zue, V. (1993). TIMIT Acoustic-Phonetic Continuous Speech Corpus LDC93S1. In (Vol. 1993): Philadelphia: Linguistic Data Consortium.
Jung, J.-W., Heo, H.-S., Yang, I.-H., Shim, H.-J., & Yu, H.-J. (2018). Avoiding speaker overfitting in end-to-end dnns using raw waveform for text-independent speaker verification. *Extraction, 8*, 23–24.
Kadiri, S. R., Prasad, R., & Yegnanarayana, B. (2020). Detection of glottal closure instant and glottal open region from speech signals using spectral flatness measure. *Speech Communication, 116*, 30–43.
Kahn, J., Audibert, N., Bonastre, J.-F., & Rossato, S. (2011). Inter and Intra-speaker Variability in French: An Analysis of Oral Vowels and Its Implication for Automatic Speaker Verification. In ICPhS (pp. 1002–1005).
Kanagasundaram, A., Vogt, R., Dean, D. B., Sridharan, S., & Mason, M. W. (2011). I-vector based speaker recognition on short utterances. In Proceedings of the 12th Annual Conference of the International Speech Communication Association (pp. 2341–2344): International Speech Communication Association (ISCA).
Kawakami, Y., Wang, L., Kai, A., & Nakagawa, S. (2014). Speaker identification by combining various vocal tract and vocal source features. In International conference on text, speech, and dialogue (pp. 382–389): Springer.
Kekre, H., Athawale, A., & Desai, M. (2011). Speaker identification using row mean vector of spectrogram. In Proceedings of the International Conference & Workshop on Emerging Trends in Technology (pp. 171–174): ACM.
Kinnunen, T. (2003). Spectral features for automatic text-independent speaker recognition. Licentiate's thesis.
Kominek, J., & Black, A. W. (2004). The CMU Arctic speech databases. In Fifth ISCA workshop on speech synthesis.
Kovalev, V., Kalinovsky, A., & Kovalev, S. (2016). Deep learning with theano, torch, caffe, tensorflow, and deeplearning4j: Which one is the best in speed and accuracy?.
Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems (pp. 1097–1105).
Krobba, A., Debyeche, M., & Selouani, S. A. (2019). Maximum entropy PLDA for robust speaker recognition under speech coding distortion. *International Journal of Speech Technology, 22*(4), 1115–1122.
Krothapalli, S. R., Yadav, J., Sarkar, S., Koolagudi, S. G., & Vuppala, A. K. (2012). Neural network based feature transformation for emotion independent speaker identification. *International Journal of Speech Technology, 15*(3), 335–349.
Larcher, A., Lee, K. A., Ma, B., & Li, H. (2014). Text-dependent speaker verification: Classifiers, databases and RSR2015. *Speech Communication, 60*, 56–77.
Larcher, A., Lee, K. A., & Meignier, S. (2016). An extensible speaker identification sidekit in python. In 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 5095–5099): IEEE.
Larsson, J. (2014). Optimizing text-independent speaker recognition using an LSTM neural network.
Lawson, A., Vabishchevich, P., Huggins, M., Ardis, P., Battles, B., & Stauffer, A. (2011). Survey and evaluation of acoustic features for speaker recognition. In 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 5444–5447): IEEE.
LeCun, Y., Bengio, Y., & Hinton, G. (2015). *Deep learning. nature, 521*(7553), 436–444.
Lecun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE, 86*(11), 2278–2324.
Leonard, L. C. (2017). Web-Based Behavioral Modeling for Continuous User Authentication (CUA). In Advances in Computers (Vol. 105, pp. 1–44): Elsevier.

Li, C., Sanchez, R.-V., Zurita, G., Cerrada, M., Cabrera, D., & Vásquez, R. E. (2015). Multimodal deep support vector classification with homologous features and its application to gearbox fault diagnosis. *Neurocomputing, 168*, 119–127.
Li, Z., & Gao, Y. (2016). Acoustic feature extraction method for robust speaker identification. *Multimedia Tools and Applications, 75*(12), 7391–7406.
Lin, Z., Feng, M., Santos, C. N. d., Yu, M., Xiang, B., Zhou, B., & Bengio, Y. (2017). A structured self-attentive sentence embedding. arXiv preprint arXiv:1703.03130.
Liu, H., & Motoda, H. (2012). *Feature selection for knowledge discovery and data mining, (Vol. 454):*. Springer Science & Business Media.
Liu, Z., Wu, Z., Li, T., Li, J., & Shen, C. (2018). GMM and CNN hybrid method for short utterance speaker recognition. *IEEE Transactions on Industrial Informatics, 14*(7), 3244–3252.
Lukic, Y., Vogt, C., Dürr, O., & Stadelmann, T. (2016). Speaker identification and clustering using convolutional neural networks. In 2016 IEEE 26th international workshop on machine learning for signal processing (MLSP) (pp. 1–6): IEEE.
Luque-Suárez, F., Camarena-Ibarrola, A., & Chávez, E. (2019). Efficient speaker identification using spectral entropy. *Multimedia Tools and Applications, 78*(12), 16803–16815.
Ma, Z., & Leijon, A. (2011). Super-Dirichlet mixture models using differential line spectral frequencies for text-independent speaker identification. In Twelfth Annual Conference of the International Speech Communication Association.
Mallat, S. (1999). *A wavelet tour of signal processing*. Elsevier.
Manikandan, K., & Chandra, E. (2016). Speaker Identification using a Novel Prosody with Fuzzy based Hierarchical Decision Tree Approach. *Indian Journal of Science and Technology, 9*, 44.
Manikandan, K. H., & Chandra, E. (2016). Speaker Identification using a Novel Prosody with Fuzzy based Hierarchical Decision Tree Approach. In *Indian Journal of Science and Technology* (p. 9).
Mannepalli, K., Sastry, P. N., & Suman, M. (2017). A novel adaptive fractional deep belief networks for speaker emotion recognition. *Alexandria Engineering Journal, 56*(4), 485–497.
Marcel, S., Nixon, M., & Li, S. (2014). Handbook of Biometric Anti-Spoofing-Trusted Biometrics under Spoofing Attacks, ser. Advances in Computer Vision and Pattern Recognition. Springer.
Matejka, P., Burget, L., Schwarz, P., & Cernocky, J. (2006). Brno university of technology system for nist 2005 language recognition evaluation. In 2006 IEEE Odyssey-The Speaker and Language Recognition Workshop (pp. 1–7): IEEE.
Mathieu, B., Essid, S., Fillon, T., Prado, J., & Richard, G. (2010). YAAFE, an Easy to Use and Efficient Audio Feature Extraction Software. In ISMIR (pp. 441–446).
McKay, C., Fujinaga, I., & Depalle, P. (2005). jAudio: A feature extraction library. In Proceedings of the International Conference on Music Information Retrieval (pp. 600–603).
Medikonda, J., & Madasu, H. (2018). Higher order information set based features for text-independent speaker identification. *International Journal of Speech Technology, 21*(3), 451–461.
Medikonda, J., & Madasu, H. J. I. J. o. S. T. (2018). Higher order information set based features for text-independent speaker identification. 21, 451–461.
Michalevsky, Y., Talmon, R., & Cohen, I. (2011). Speaker identification using diffusion maps. In 2011 19th European signal processing conference (pp. 1299–1302): IEEE.
MicroPyramid. (2011). Understanding Audio Quality: Bit Rate, Sample Rate. In (Vol. 2011). MicroPyramid blog.
Moffat, D., Ronan, D., & Reiss, J. D. (2015). An evaluation of audio feature extraction toolboxes.
Mokgonyane, T. B., Sefara, T. J., Manamela, M. J., & Modipa, T. I. (2019). The Effects of Data Size on Text-Independent Automatic Speaker Identification System. In 2019 International Conference on Advances in Big Data, Computing and Data Communication Systems (icABCD) (pp. 1-6): IEEE.
Morrison, G. S., Sahito, F. H., Jardine, G., Djokic, D., Clavet, S., Berghs, S., & Goemans Dorny, C. (2016). INTERPOL survey of the use of speaker identification by law enforcement agencies. *Forensic Science International, 263*, 92–100.
Mporas, I., Safavi, S., Gan, H. C., & Sotudeh, R. (2016). Evaluation of classification algorithms for text dependent and text independent speaker identification. In: IEICE.
Mujtaba, G., Shuib, L., Idris, N., Hoo, W. L., Raj, R. G., Khowaja, K., Shaikh, K., & Nweke, H. F. (2019). Clinical text classification research trends: systematic literature review and open issues. Expert Systems with Applications, 116, 494–520.
Nagori, V. (2016). Fine tuning the parameters of back propagation algorithm for optimum learning performance. In 2016 2nd International Conference on Contemporary Computing and Informatics (IC3I) (pp. 7–12): IEEE.
Nagrani, A., Chung, J. S., & Zisserman, A. (2017). Voxceleb: a large-scale speaker identification dataset. arXiv preprint arXiv:1706.08612.
Naik, J., & Doddington, G. (1987). Evaluation of a high performance speaker verification system for access Control. In Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'87. (Vol. 12, pp. 2392–2395): IEEE.
Nair, V., & Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In Proceedings of the 27th international conference on machine learning (ICML-10) (pp. 807–814).
Nakamura, A. (2002). Acoustic modeling for speech recognition based on a generalized Laplacian mixture distribution. *Electronics and Communications in Japan (Part II: Electronics), 85*(11), 32–42.
Nemer, E., Goubran, R., & Mahmoud, S. (2001). Robust voice activity detection using higher-order statistics in the LPC residual domain. *IEEE Transactions on Speech and Audio Processing, 9*(3), 217–231.
Ng, A. Y., & Jordan, M. I. (2002). On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In Advances in neural information processing systems (pp. 841–848).
Nosratighods, M., Ambikairajah, E., Epps, J., & Carey, M. J. (2010). A segment selection technique for speaker verification. *Speech Communication, 52*(9), 753–761.
Novotný, O., Plchot, O., Glembek, O., Černocký, J.“., & Burget, L. (2019). Analysis of DNN Speech Signal Enhancement for Robust Speaker Recognition. *Computer Speech & Language, 58*, 403–421.
Nweke, H. F., Teh, Y. W., Mujtaba, G., & Al-garadi, M. A. (2019). Data fusion and multiple classifier systems for human activity detection and health monitoring: Review and open research directions. *Information Fusion, 46*, 147–170.
Nweke, H. F., Teh, Y. W., Mujtaba, G., Alo, U. R., & Al-garadi, M. A. (2019). Multi-sensor fusion based on multiple classifier systems for human activity identification. *Human-centric Computing and Information Sciences, 9*, 34.
Ouyang, Z., Sun, X., Chen, J., Yue, D., & Zhang, T. (2018). Multi-view stacking ensemble for power consumption anomaly detection in the context of industrial internet of things. *IEEE Access, 6*, 9623–9631.
Palm, R. (2014). Deeplearntoolbox, a matlab toolbox for deep learning. Online]. Disponível em: https://github. com/rasmusbergpalm/DeepLearnToolbox.
Panayotov, V., Chen, G., Povey, D., & Khudanpur, S. (2015). Librispeech: an ASR corpus based on public domain audio books. In 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 5206–5210): IEEE.
Petrovska, D., Hennebert, J., Melin, H., & Genoud, D. (1998). Polycost: a telephone-speech database for speaker recognition. Proc. RLA2C, Avignon, France, 211–214.
Petry, A., & Barone, D. A. C. (2002). Speaker identification using nonlinear dynamical features. *Chaos, Solitons & Fractals, 13*(2), 221–231.
Picone, J. W. (1993). Signal modeling techniques in speech recognition. *Proceedings of the IEEE, 81*(9), 1215–1247.
Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlicek, P., Qian, Y., & Schwarz, P. (2011). The Kaldi speech recognition toolkit. In IEEE 2011 workshop on automatic speech recognition and understanding: IEEE Signal Processing Society.
Quinlan, J. R. (1986). Induction of decision trees. Machine learning, 1, 81–106.
Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE, 77*(2), 257–286.
Ralph Abbey, T. H., & Tao Wang. (2017). Methods of Multinomial Classification Using Support Vector Machines In SAS® Global Forum. Orlando, Florida: SAS Institute Inc.
Renisha, G., & Jayasree, T. (2019). Cascaded Feedforward Neural Networks for speaker identification using Perceptual Wavelet based Cepstral Coefficients. *Journal of Intelligent & Fuzzy Systems, 37*(1), 1141–1153.
Revathi, A., & Venkataramani, Y. (2009). Text independent composite speaker identification/verification using multiple features. In 2009 WRI World congress on computer science and information engineering (Vol. 7, pp. 257–261): IEEE.
Reynolds, D. A. (2002). An overview of automatic speaker recognition technology. In 2002 IEEE International Conference on Acoustics, Speech, and Signal Processing (Vol. 4, pp. IV-4072-IV-4075): IEEE.
Reynolds, D. A., & Rose, R. C. (1995). Robust text-independent speaker identification using Gaussian mixture speaker models. *IEEE Transactions on Speech and Audio Processing, 3*(1), 72–83.
Sadıç, S., & Bilginer Gülmezoğlu, M. (2011). Common vector approach and its combination with GMM for text-independent speaker recognition. *Expert Systems with Applications, 38*(9), 11394–11400.
Saha, G., Chakroborty, S., & Senapati, S. (2005). A new silence removal and endpoint detection algorithm for speech and speaker recognition applications. In Proceedings of the 11th national conference on communications (NCC) (pp. 291–295).
Sahoo, T. R., & Patra, S. (2014). Silence Removal and Endpoint Detection of Speech Signal for Text Independent Speaker Identification. *International Journal of Image, Graphics & Signal Processing, 6*.
Saquib, Z., Salam, N., Nair, R. P., Pandey, N., & Joshi, A. (2010). A survey on automatic speaker recognition systems. In Signal Processing and Multimedia (pp. 134–145): Springer.
Sardar, V., & Shirbahadurkar, S. (2018a). Speaker Identification of Whispering Sound: Effect of Different Features on the Identification Accuracy. International Journal of Pure and Applied Mathematics, 118.
Sardar, V. M., & Shirbahadurkar, S. D. (2018). Speaker identification of whispering speech: An investigation on selected timbrel features and KNN distance measures. *International Journal of Speech Technology, 21*(3), 545–553.
Sardar, V. M., & Shirbahadurkar, S. (2019). Timbre features for speaker identification of whispering speech: Selection of optimal audio descriptors. *International Journal of Computers and Applications*, 1–7.
Sarma, M., & Sarma, K. K. (2013). Vowel phoneme segmentation for speaker identification using an ANN-based framework. *Journal of Intelligent Systems, 22*, 111–130.
Schmandt, C., & Arons, B. (1984). A conversational telephone messaging system. *IEEE Transactions on Consumer Electronics, CE-30*(3), xxi–xxiv.
Seide, F., & Agarwal, A. (2016). CNTK: Microsoft's open-source deep-learning toolkit. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 2135–2135): ACM.
Sekkate, S., Khalil, M., & Adib, A. (2019). Speaker identification for OFDM-based aeronautical communication system. *Circuits, Systems, and Signal Processing, 38*(8), 3743–3761.
Shah, J. K., Smolenski, B. Y., Yantorno, R. E., & Iyer, A. N. (2004). Sequential k-nearest neighbor pattern recognition for usable speech classification. In 2004 12th European Signal Processing Conference (pp. 741–744): IEEE.
Shahamiri, S. R., & Salim, S. S. B. (2014). A multi-views multi-learners approach towards dysarthric speech recognition using multi-nets artificial neural networks. *IEEE Transactions on Neural Systems and Rehabilitation Engineering, 22*(5), 1053–1063.

Shahin, I., Nassif, A. B., & Hamsa, S. (2020). Novel cascaded Gaussian mixture model-deep neural network classifier for speaker identification in emotional talking environments. *Neural Computing and Applications, 32*(7), 2575–2587.
Shahin, M. A., Epps, J., & Ahmed, B. (2016). Automatic Classification of Lexical Stress in English and Arabic Languages Using Deep Learning. In INTERSPEECH (pp. 175–179).
Shannon, C.E. (2001). A mathematical theory of communication. 5, 3–55.
Shi, Y., Huang, Q., & Hain, T. (2020). Weakly Supervised Training of Hierarchical Attention Networks for Speaker Identification. arXiv preprint arXiv:2005.07817.
Shlens, J. (2014). A tutorial on principal component analysis. arXiv preprint arXiv:1404.1100.
Shrestha, A., & Mahmood, A. (2019). Review of deep learning algorithms and architectures. *IEEE Access, 7*, 53040–53065.
Siam, A. I., El-khobby, H. A., Elnaby, M. M. A., Abdelkader, H. S., & El-Samie, F. E. A. (2019). A novel speech enhancement method using Fourier series decomposition and spectral subtraction for robust speaker identification. *Wireless Personal Communications, 108*(2), 1055–1068.
Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.
Sohn, J., Kim, N. S., & Sung, W. J. I. s. p. l. (1999). A statistical model-based voice activity detection. 6, 1–3.
Soleymanpour, M., & Marvi, H. (2017). Text-independent speaker identification based on selection of the most similar feature vectors. *International Journal of Speech Technology, 20*(1), 99–108.
Stolar, M. N., Lech, M., Bolia, R. S., & Skinner, M. (2017). Real time speech emotion recognition using RGB image classification and transfer learning. In 2017 11th International Conference on Signal Processing and Communication Systems (ICSPCS) (pp. 1–8).
Sun, L., Chen, J., Xie, K., & Gu, T. (2018). Deep and shallow features fusion based on deep convolutional neural network for speech emotion recognition. *International Journal of Speech Technology, 21*(4), 931–940.
Sun, L., Gu, T., Xie, K., & Chen, J. (2019). Text-independent speaker identification based on deep Gaussian correlation supervector. *International Journal of Speech Technology, 22*(2), 449–457.
Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. In Advances in neural information processing systems (pp. 3104–3112).
Suykens, J. A., & Vandewalle, J. (1999). Least squares support vector machine classifiers. *Neural Processing Letters, 9*, 293–300.
Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2015). Going deeper with convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1–9).
Team, D. (2016). Deeplearning4j: Open-source distributed deep learning for the jvm. Apache Software Foundation License, 2.
Tesauro, G. (1992). Practical issues in temporal difference learning. In Advances in neural information processing systems (pp. 259–266).
Thoman, C. (2009). Model-Based Classification of Speech Audio: Florida Atlantic University.
Tian, G., Xia, Y., Zhang, Y., & Feng, D. (2011). Hybrid genetic and variational expectation-maximization algorithm for Gaussian-mixture-model-based brain MR image segmentation. *IEEE Transactions on Information Technology in Biomedicine, 15*, 373–380.
Tirumala, S. S., & Shahamiri, S. R. (2016). A review on Deep Learning approaches in Speaker Identification. In Proceedings of the 8th international conference on signal processing systems (pp. 142–147): ACM.
Tirumala, S. S., & Shahamiri, S. R. (2017). A deep autoencoder approach for speaker identification. In Proceedings of the 9th International Conference on Signal Processing Systems (pp. 175–179): ACM.
Tirumala, S. S., Shahamiri, S. R., Garhwal, A. S., & Wang, R. (2017). Speaker identification features extraction methods: A systematic review. *Expert Systems with Applications, 90*, 250–271.
Tiwari, V., Hashmi, M. F., Keskar, A., & Shivaprakash, N. C. (2020). Virtual home assistant for voice based controlling and scheduling with short speech speaker identification. *Multimedia Tools and Applications, 79*(7-8), 5243–5268.
Togneri, R., & Pullella, D. (2011). An overview of speaker identification: Accuracy and robustness issues. *IEEE Circuits and Systems Magazine, 11*(2), 23–61.
Tokui, S., Oono, K., Hido, S., & Clayton, J. (2015). Chainer: a next-generation open source framework for deep learning. In Proceedings of workshop on machine learning systems (LearningSys) in the twenty-ninth annual conference on neural information processing systems (NIPS) (Vol. 5, pp. 1–6).
Tran, V.-T., & Tsai, W.-H. (2020). Speaker Identification in Multi-Talker Overlapping Speech Using Neural Networks. IEEE Access.
Vasilev, I. (2019). Python deep learning: exploring deep learning techniques and neural network architectures with PyTorch, Keras, and TensorFlow.
Vedaldi, A., & Lenc, K. (2015). Matconvnet: Convolutional neural networks for matlab. In Proceedings of the 23rd ACM international conference on Multimedia (pp. 689–692): ACM.
Verma, G. K. (2011). Multi-feature fusion for closed set text independent speaker identification. In International conference on information intelligence, systems, technology and management (pp. 170–179): Springer.
Vetterli, M., & Kovačevic, J. (1995). Wavelets and subband coding: Prentice-Hall, Inc.
Vogt, R. J., Lustri, C. J., & Sridharan, S. (2008). Factor analysis modelling for speaker verification with short utterances.
Wang, C. (2020). Speech Emotion Recognition Based on Multi-feature and Multi-lingual Fusion. arXiv preprint arXiv:2001.05908.
Wang, D., & Zhang, X. (2015). Thchs-30: A free chinese speech corpus. arXiv preprint arXiv:1512.01882.
Wang, J.-C., Chin, Y.-H., Hsieh, W.-C., Lin, C.-H., Chen, Y.-R., & Siahaan, E. (2015). Speaker identification with whispered speech for the access control system. *IEEE Transactions on Automation Science and Engineering, 12*(4), 1191–1199.
Wang, X., Xue, F., Wang, W., & Liu, A. (2020). A network model of speaker identification with new feature extraction methods and asymmetric BLSTM. *Neurocomputing, 403*, 167–181.
Weninger, F., Ringeval, F., Marchi, E., & Schuller, B. W. (2016). Discriminatively Trained Recurrent Neural Networks for Continuous Dimensional Emotion Recognition from Audio. In IJCAI (Vol. 2016, pp. 219–2202).
Wu, J.-D., & Lin, B.-F. (2009a). Speaker identification based on the frame linear predictive coding spectrum technique. *Expert Systems with Applications, 36*(4), 8056–8063.
Wu, J.-D., & Lin, B.-F. (2009b). Speaker identification using discrete wavelet packet transform technique with irregular decomposition. *Expert Systems with Applications, 36*(2), 3136–3143.
Wu, J.-D., & Tsai, Y.-J. (2011). Speaker identification system using empirical mode decomposition and an artificial neural network. *Expert Systems with Applications, 38*(5), 6112–6117.
Yadav, S., & Rai, A. (2018). Learning Discriminative Features for Speaker Identification and Verification. In Interspeech (pp. 2237–2241).
Yakovenko, A., & Malychina, G. (2016). Text-independent speaker recognition using radial basis function network. In International Symposium on Neural Networks (pp. 74–81): Springer.
Young, S. J., & Young, S. (1993). The HTK hidden Markov model toolkit: Design and philosophy: University of Cambridge, Department of Engineering Cambridge, England.
Yue, Y., & Yang, Y. (2020). Mobile intelligent terminal speaker identification for real-time monitoring system of sports training. *Evolutionary Intelligence*, 1–12.
Zhang, C., Koishida, K., & Hansen, J. H. (2018a). Text-independent speaker verification based on triplet convolutional neural network embeddings. IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP), 26, 1633–1644.
Zhang, C., Koishida, K., & Hansen, J. H. L. (2018b). Text-independent speaker verification based on triplet convolutional neural network embeddings. *IEEE/ACM Transactions on Audio, Speech, and Language Processing, 26*(9), 1633–1644.
Zhang, C., Patras, P., & Haddadi, H. (2019). Deep learning in mobile and wireless networking: A survey. *IEEE Communications Surveys & Tutorials, 21*(3), 2224–2287.
Zhang, T., Shao, Y., Wu, Y., Geng, Y., & Fan, L. (2020). An overview of speech endpoint detection algorithms. *Applied Acoustics, 160*, 107133. https://doi.org/10.1016/j.apacoust.2019.107133
Zhang, X., Zou, X., Sun, M., & Wu, P. (2018). Robust Speaker Recognition Using Improved GFCC and Adaptive Feature Selection. In International Conference on Security with Intelligent Computing and Big-data Services (pp. 159–169): Springer.
Zhang, Z., Wang, L., Kai, A., Yamada, T., Li, W., & Iwahashi, M. (2015). Deep neural network-based bottleneck feature and denoising autoencoder-based dereverberation for distant-talking speaker identification. EURASIP Journal on Audio, Speech, and Music Processing, 2015, 12.
Zhao, X., & Wang, D. (2013). Analyzing noise robustness of MFCC and GFCC features in speaker identification. In 2013 IEEE international conference on acoustics, speech and signal processing (pp. 7204–7208): IEEE.
Zhao, X., Wang, Y., & Wang, D. (2014). Robust speaker identification in noisy and reverberant conditions. *IEEE/ACM Transactions on Audio, Speech, and Language Processing, 22*(4), 836–845.