[1]Ankumon Sarmah

[2] Rizwan Rehman

[3] Priyakshi Mahanta

[4] Kankana Dutta

[5] Sumpi Saikia

[6] Kimasha Borah

A Novel Approach for Automatic Speaker Identification and Word Identification of Tai-Phake Speakers from short-duration spoken words using MMCCT and Feed-Forward Neural Network

Journal of Electrical Systems

***Abstract:* - Objective**: Speaker identification and word identification are critical tasks in language processing and artificial intelligence. This research focuses on applying feed-forward neural networks to the specific context of the Tai-Phake language, an endangered language spoken in parts of Northeast India. The primary objective is to develop effective models for both speaker identification and word identification in Tai-Phake, leveraging the capabilities of neural networks. **Methods**: The study begins by collecting a corpus of Tai-Phake speech data from 18 speakers recording 50 different words 10 times each, which is annotated and pre-processed to facilitate model training. For speaker recognition, a feed-forward neural network architecture is designed to accurately identify and authenticate individuals based on their unique vocal characteristics. The model's performance is evaluated using metrics such as accuracy, precision, recall, and F1 score. Additionally, for word recognition, another neural network model is developed to accurately identify spoken Tai-Phake utterances. This involves training the model on a labeled dataset of spoken words, optimizing it for robust performance across various dialectal variations and speaking styles within the Tai-Phake community. **Findings**: Experimental results demonstrate the effectiveness of the proposed neural network models in achieving high accuracy rates for both speaker and word recognition tasks in Tai-Phake. The implications of this research extend to the preservation and documentation of endangered languages, showcasing how advanced machine learning techniques can contribute to linguistic research and cultural heritage preservation efforts. **Novelty**: No prior work has been done in Tai-phake speaker identification and word identification using a combination of Feature Fusion and Neural Networks.

Keywords: Speaker identification, word identification, neural networks, Tai Phake language, endangered languages.

I. INTRODUCTION

The use of voice to connect with gadgets has grown significantly over the past few years, ranking among the most popular forms of interaction. The rich and valuable information that speech signals constantly provide, includes a speaker's accent, gender, emotion, and other distinctive qualities. Speaker recognition (SR) is the process of recognizing a person by his or her distinctive voice. Speaker identification and speaker verification are two of the main problems that speaker recognition addresses. In speaker identification, a speech produced by an unidentified speaker is examined and contrasted with speech models of recognized speakers. The speaker who matches the input utterance model the closest is identified as the unknown speaker.

Classification and feature extraction are the two main phases in the speaker identification process. A collection of attributes dependent on the user is extracted during the feature extraction phase. It varies from speaker to speaker. The process of classification involves labeling each speaker using the collected features by a trained classifier, such as an Artificial Neural Network (ANN), GMM, or Support Vector Machine (SVM). Speaker identification can be text-independent or text-dependent[1]. The system knows beforehand about the spoken utterances for the text-dependent class. In contrast, a text-independent class is unaffected by the relevant context. This paper presents speaker identification in both text-dependent and text-independent cases. Both training and testing (identification) are phases of a Speaker Identification system [2]. A model is created for each speaker during the training phase, and a decision is reached regarding the identity of an unknown speaker during the testing phase after comparing it to the stored models.

Word recognition will be used to describe the computational procedures that listeners utilize to recognize spoken words in their acoustic-phonetic and phonological forms [3]. One way to conceptualize word recognition is as a

*Corresponding author: Ankumon Sarmah (ankumonsarmah2009@gmail.com)

[1,2,4,5,6] Author Affiliation: Centre for Computer Science and Applications, Dibrugarh University

[3] Author Affiliation: Jorhat Engineering College, Jorhat

type of pattern recognition. It is considered that word recognition relies on the same sensory and perceptual processes regardless of whether the input contains words or pronounceable nonwords.

Tai Phake Language, a low-resource language in Assam, is the major dataset that we took into consideration for our experiment. From the prehistoric era, Assam, a state in northeastern India, has been a multilingual and diverse state with distinct languages and cultures spoken by each of its communities. Over a million people identify as Tai, and the majority of them live in Northeast India. However, Tai Ahoms, who no longer speak Tai, make up more than 99% of them. Other tribes that identify as Tai, in addition to the Ahoms, include the Aiton, Khamti, Khamyang, Nora, Phake, and Turung. All those places have varying degrees of Tai speaking [4]. However, there are a number of reasons why these seven communities have given up on their native tongue, including a lack of funding for language instruction, the influence of other languages, and a lack of backing from governmental and non-governmental organizations. There are, nevertheless, eleven Phake villages in Assam and Arunachal Pradesh. It is possible that 2,000 people speak it, and young people are still picking up the language [4].

The remaining sections of the manuscript are organized as follows: Section II offers a review of the literature on speaker recognition, including identification and verification tasks and pertinent citations to previous works; Section III presents the methodology along with the dataset that was selected; Section IV presents and discusses the results; and finally, Section V offers discussions and recommendations for further research.

II. Literature Review

Speaker recognition—the process of recognizing or authenticating someone based on the features of their voice—has attracted a lot of attention from academics and business leaders because of its many uses in security systems, personalized services, and human-computer interaction. The 1990s and early 2000s saw the application of Multilayer Perceptrons (MLPs), or feedforward neural networks, to speech recognition. These networks, though innovative, were limited by their shallow architectures and struggled with sequential data. The introduction of Recurrent Neural Networks (RNNs) in the early 2000s allowed for better handling of sequential data, as RNNs can maintain a form of memory about previous inputs. However, traditional RNNs faced challenges with long-range dependencies due to vanishing and exploding gradient problems.

Deep Embeddings for speaker identification emerged in the 2010s. Techniques such as x-vectors and d-vectors utilize deep learning models to generate compact representations of a speaker's voice, which are used for tasks like speaker verification and identification. These embeddings capture unique vocal characteristics, making them highly effective for distinguishing between speakers. Challenges such as noise robustness, low-resource languages, and real-time processing continue to drive research in the field. Ensuring privacy and security of voice data also remains a significant concern as voice recognition systems become more pervasive. Addressing these issues will be crucial for the continued advancement and adoption of machine learning methods in speech and speaker recognition.

To improve speaker recognition performance, this paper offers a novel method that combines deep neural networks (DNNs) with conventional approaches like i-vector and Probabilistic Linear Discriminant Analysis (PLDA)[5]. The article under evaluation belongs to the hybrid technique category, using i-vector and PLDA as inspiration for modeling speaker variability and discriminative training while utilizing DNNs for feature learning. The suggested method tries to retain efficiency and scalability while improving speaker detection performance by integrating various strategies. In summary, a possible path toward raising the bar for speaker recognition system technology is the combination of deep learning and conventional approaches.

The study by David Snyder et al. focuses on speaker verification, a crucial task in speaker recognition systems, where the objective is to confirm a speaker's stated identification based on speech traits without the need for particular text instructions [6]. By putting out a deep neural network-based strategy for text-independent speaker verification and concentrating on directly learning speaker embeddings from raw audio signals, the study under review adds to the body of knowledge. The suggested approach seeks to increase performance in real-world circumstances and get beyond the drawbacks of conventional speaker verification techniques by utilizing the representational capability of deep neural networks. To sum up, deep neural network embeddings are a promising path for improving the state-of-the-art in speaker verification that is independent of text. The work offers insightful information about the planning and execution of deep learning-based speaker verification systems, laying the groundwork for future studies and advancements in this field.

The lack of labeled training data for robust model construction is a major challenge in speaker recognition systems, particularly in situations where each speaker has limited data available [7]. The study under evaluation adds to the body of research by putting forth techniques designed especially for low-data automatic speaker detection. The suggested method seeks to improve speaker identification systems' robustness and generalization by tackling the problems caused by data scarcity, especially in real-world scenarios where gathering sizable labeled datasets may be difficult or impossible. To sum up, automatic speaker recognition with little data is a serious issue that affects how speaker identification systems are used in real-world scenarios.

The paper explains a deep learning method for speaker identification, a task in speaker recognition where the goal is to identify a speaker from an audio sample available [8]. The study under evaluation adds to the body of knowledge by putting forth a deep neural network model created especially for challenges involving speaker recognition. With the potential to address issues with robustness and scalability, the suggested model seeks to attain state-of-the-art performance in speaker recognition by utilizing the representational capacity of deep learning and advances in neural network topologies. The paper provides valuable insights into the design and implementation of deep learning-based speaker identification systems, offering a foundation for further research and development in this area.

A key element of many applications, such as forensic investigation, security systems, and tailored services, is speaker recognition. Deep learning methods have significantly improved accuracy and robustness in speaker detection, revolutionizing the discipline in recent years. This article gives a summary of the main developments, approaches, and difficulties in deep learning-based speaker recognition[9]. Because deep learning makes use of enormous datasets and intricate neural network topologies, advanced speech recognition systems have been made possible. For speaker feature extraction and modeling, Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and their variations have been widely used. Hybrid architectures combining CNNs and RNNs have been proposed to leverage both spatial and temporal information in speech data. These architectures have shown promising results in speaker verification and identification tasks.

The authors of the paper, Jahangir, Rashid, et al. presented a novel method for text-independent speaker identification by utilizing deep neural networks and feature fusion [10]. The technique uses a deep neural network design to combine data from many acoustic features and representations, making it possible to automatically learn discriminative features from unprocessed voice signals. The method aims to increase speaker identification systems' accuracy and resilience by taking advantage of the complementing qualities of several variables, especially in difficult real-world situations. The body of research shows that deep learning and feature fusion methods are useful for text-independent speaker identification. By combining these approaches, they aim to advance the state-of-the-art in speaker identification technology and contribute to the development of more reliable and versatile systems for real-world applications.

Because speaker identification, a branch of speaker recognition, has so many uses—including security, forensics, personalized services, and human-computer interaction—it has attracted a lot of attention in academic and industry settings. Researchers have been using artificial intelligence (AI) approaches more and more to improve speaker recognition systems' robustness, efficiency, and accuracy as these techniques have improved. The writers of this paper [11]. The authors provides an extensive analysis of this field, highlighting the most recent developments, obstacles, and potential paths forward. They discuss the strengths and limitations of different AI approaches, highlight key research findings, and identify areas for further investigation. By synthesizing existing literature and outlining research challenges, the authors aim to provide insights into the current state-of-the-art in AI-based speaker identification and inspire future research efforts to address emerging challenges and opportunities.

Many years of study have been devoted to text-independent speaker identification, an essential component of many security and authentication systems. The thorough examination of feature extraction methods and their historical development is one of the main contributions of the paper by Kinnunen, Tomi, and Haizhou Li. [12]. The study describes several feature extraction techniques, such as shifted delta cepstral (SDC) features and perceptual linear prediction (PLP), two more recent developments in addition to the previously established cepstral-based features. The research offers important insights into the design concerns for text-independent speaker recognition systems by outlining the advantages and disadvantages of each strategy.

It has long been known that Mel-Frequency Cepstral Coefficients (MFCCs) are useful features for speech signal representation in ASR systems. The MFCC-based approach to ASR is described in detail in this paper and includes steps for feature extraction, model training, and speaker identification or verification [13]. The benefits and drawbacks of employing MFCCs for ASR are discussed in the paper, with an emphasis on how well they capture speaker-specific information while recognizing possible drawbacks including sensitivity to channel effects and speaking style variations. The authors also discuss techniques like feature normalization, dimensionality reduction, and model ensembling for enhancing ASR performance. Furthermore, the paper addresses the importance of dataset selection and model evaluation in ASR research. Large-scale annotated datasets are essential for training machine learning models effectively and ensuring their generalization to unseen speakers and conditions

The paper addresses a critical challenge in speaker verification systems: handling short-duration speech segments while maintaining high accuracy and robustness [14]. The paper proposes a discriminative neural embedding learning framework for short-duration text-independent speaker verification. The key idea is to learn speaker embeddings directly from raw speech signals using deep neural networks, which can capture high-level speaker-specific information while being robust to variations in speech duration and content. One of the main contributions of the paper is its exploration of various neural network architectures and training strategies for speaker embedding learning. The effectiveness of the proposed approach is demonstrated through comprehensive experiments on benchmark speaker verification datasets, including those with short-duration utterances. The results show that the proposed discriminative neural embedding learning framework outperforms traditional methods and achieves state-of-the-art performance in short-duration text-independent speaker verification tasks The results show that the proposed discriminative neural embedding learning framework outperforms traditional methods and achieves state-of-the-art performance in short-duration text-independent speaker verification tasks.

III. Proposed Methodology

In this section, we present a comparison of two methods for speaker identification as well as word identification from a recorded speech sample. For the testing, voice samples from 18 speakers were first collected. After that, a feature vector was produced by merging several helpful features that were obtained from the compiled speaker utterances, including MFCC, Mel, Chroma, Contrast, Tonnetz, and duration. This feature vector was fed into two feed-forward deep neural network architectures to generate the speaker recognition model and word identification accordingly. The suggested work's experimental outcome is shown in terms of speaker identification accuracy and word identification accuracy. It has been determined that the suggested strategy is effective in terms of accuracy value, recall, f1 score, and precision. Finally, an alternative test set was used to evaluate the performance of the developed model. The block diagram of the proposed methodology is shown in Figure 1.

A. Dataset

A key component of both Speaker and Word identification system is the dataset [15]. Many systems for identifying and recognizing speakers just require the spoken word of the registered group of users. To identify the speaker in our investigation, we have selected the Tai-Phake language as the command to enter into the model. A total of fifty words from the Tai-Phake language, as spoken by native speakers, have been selected for our study. The dataset used in this work includes 18 speakers (10 female and 8 male). Nine thousand voice samples were produced by recording each syllable ten times by the speakers. For low-resource languages, many works have been done employing extremely few speakers, ranging from 06 to 30, for speaker identification [15][16][17][18][19][20]. Speaker information of the dataset used is shown in Table 1 and the time-amplitude plot of twelve different word samples is shown in Fig 1.

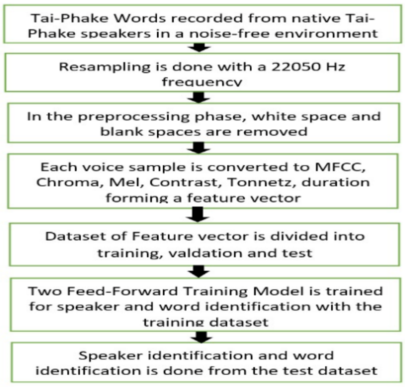
Tai-Phake Words recorded from native Tai-Phake speakers in a noise-free environment
Resampling is done with a 22050 Hz frequency
In the preprocessing phase, white space and blank spaces are removed
Each voice sample is converted to MFCC, Chroma, Mel, Contrast, Tonnetz, duration forming a feature vector
Dataset of Feature vector is divided into training, valdation and test
Two Feed-Forward Training Model is trained for speaker and word identification with the training dataset
Speaker identification and word identification is done from the test dataset

Figure 1: Block Diagram for both Speaker and Word Identification System

B. Speech Pre-processing

In systems where silence or background noise is inappropriate, pre-processing speech signals is an essential step. Systems like automatic speaker identification and word identification, where most spoken words contain features associated with the speaker, require efficient feature extraction algorithms from speech signals. As a result, every speaker's voice sample that was recorded was captured in a quiet setting. To extract features, the blank space preceding and after each speech sample is eliminated, leaving only the speaker's voice.

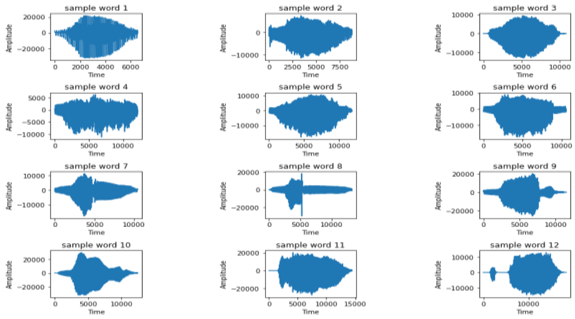

Figure 2: Figure represents the graphical representation of 12 Tai-Phake words spectrogram

C. Feature Engineering

Feature engineering is the process of selecting, altering, and transforming raw data into features that may be used in supervised learning. Better features may need to be developed and trained before machine learning may be applied to new tasks. Any quantifiable input that can be employed in a predictive model is called a feature; voice characteristics or the form of an object's body are two examples. Thus, the act of using statistical or machine learning methods to convert raw data into desired features is known as feature engineering. A neural network model needs to be trained using feature engineering. The quality of a feature collection typically determines how well a classification performs. Therefore, fewer accurate categorization results may arise from irrelevant features. Finding discriminative feature sets is a critical problem in deep learning and machine learning to achieve satisfactory classification performance. The caliber of the features employed in the classification task has a major impact on whether a speaker identification model succeeds or fails. Consequently, in machine learning and deep learning, feature engineering is an essential step [21]. If there is a strong link between the class and the retrieved attributes, classifying the data will be straightforward and accurate. On the other hand, if there is a poor correlation between the retrieved features and the class, the classification procedure will be difficult and inaccurate [21]. We have extracted a collection of features from various voice samples in this study, including MFCC, Mel, Chroma, Contrast, Tonnetz, and duration. Following the feature selection process, the features are joined to create a vector, which is subsequently supplied to the neural network model for training.

Table 1: Speaker information of the datasets used

Sl. No.	Speakers	Gender	Age
1	SP1	Male	54
2	SP2	Male	45
3	SP3	Female	30
4	SP4	Female	36
5	SP5	Male	26
6	SP6	Male	25
7	SP7	Male	21
8	SP8	Female	20
9	SP9	Female	21
10	SP10	Female	22
11	SP11	Female	24
12	SP12	Male	23
13	SP13	Male	28
14	SP14	Female	25
15	SP15	Male	50
16	SP16	Female	26
17	SP17	Female	27
18	SP18	Female	18

a. Proposed Feature Set

The short-period power spectrum of a sound wave is represented by the Mel-Frequency Cepstrum (MFC); the collection of MFC coefficients is called the Mel frequency cepstral coefficient (MFCC), and it is based on human auditory characteristics [22]. Because its coefficients are based on human hearing perceptions, MFCC is a feature extraction method that is frequently employed in automatic speaker identification [16]. For our investigation, a total of forty MFCC coefficients have been extracted. The Mel-scaled spectrogram was another feature we took out of the voice sample. The Mel spectrogram's 128 coefficients are used in this study. Similar to this, the we have extracted (chromagram) Chroma, Contrast (spectral contrast), and Tonnetz (tonal centroid characteristics), yielding 12, 7, and 6 coefficients, respectively. In our study, we have proposed two different Neural network models and compared their efficiency in terms of accuracy. Both models have all five features MFCC, Mel, Chroma, Contrast, and Tonnetz which form a total of 193 components. For the word identification model along with these features we have added duration as another feature forming 194 components. A matrix of order *mxn* is returned by the mfcc() function, where *n* is set to 40 to return 40 coefficients and *m* is dependent on the length of the voice sample. Every column's mean is determined, and a 1-dimensional vector containing 40 characteristics is saved in the variable *mfcc*. Additionally, the melspectrogram() function provided a matrix of *mxn*, where *n* is 128 and *m* is dependent on the length of the voice sample. Once more, a vector of 128 characteristics is kept in a variable called *mel* after mean is determined for each column. A matrix of *mxn* was also given by the chromastft () function, where *m* is dependent on the duration of the voice sample and *n* is 12. A vector of 12 features is saved in a variable called *chroma* after the mean is once more calculated for each column. Additionally, the spectral contrast() function returned a matrix of size *mxn*, where *n* is 7 and *m* is dependent on the duration of the voice sample. Once more, the mean is determined for each column, and a vector of seven features is saved in a variable called *contrast*. Additionally, the tonnetz () method provided a matrix of *mxn*, where *n* is 6 and *m* is dependent on the duration of the voice sample. Once more, a vector of six features is saved in a variable called *tonnetz* after the mean is calculated for each column.

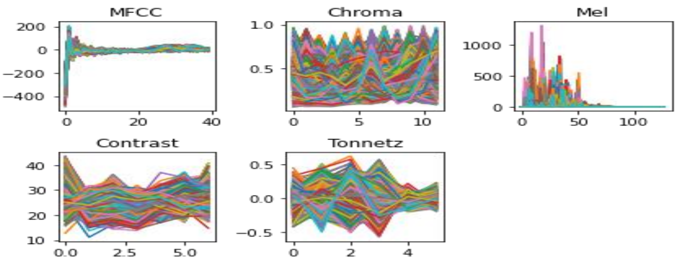

Figure 3: images show the graphical representation of the respective feature matrix of various spectrogram where x coordinates represent the number of features and y coordinates represents their magnitude.

b. Algorithm for Feature Extraction

Algorithm 1 MMCCT(MFCC, Mel, Chroma, Contrast, Tonnetz) Features of Speaker Voice Sample

```
Input : path to the folder of Speaker Voice Sample
For all file in voice sample folder
    a←GetMFCCFeatures(file)
    mfcc[] ←mean(a)
    b←GetMelFeatures(file)
    mel[] ←mean(b)
    c←GetChromaFeatures(file)
    chroma[] ←mean(c)
    d←GetContrastFeatures(file)
    contrast[]←mean(d)
    e←GetTonnetzFeatures(file)
    tonnetz[]←mean(e)
    duration ← Get the duration o
    speaker_name←GetNameofSpeaker(file)
    mmcct[]←append(mfcc[],mel[],chroma[],contrast[],tonnetz[],duration,speaker_name)
end
```

D. SPEAKER AND WORD IDENTIFICATION MODEL

The functioning of the human brain is represented by neural networks. They make it possible for computer programs to see patterns and fix common machine-learning problems. The procedure of Automatic Speaker Identification (ASI) involves matching a speaker's speech sample with their previously recorded voice in order to automatically identify the speaker. For ASI, the machine learning strategy has grown in favor in recent years. Convolutional neural networks (CNN) [27,28,29], deep neural networks (DNN) [23,24,25,26], and artificial neural networks (ANN) [30,31] are some of the machine learning techniques that ASI has employed recently.

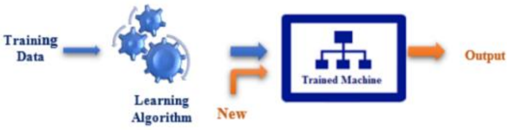

Figure 4: Machine Learning process [20]

In this paper, we have presented and contrasted two distinct feed-forward neural network models for each word and speaker identification. Furthermore, these two feed-forward deep neural networks' total performance was comparable to that of conventional classification algorithms. A brief explanation of our suggested deep neural network model is given in the paragraph that follows.

a. Proposed Feed Forward Neural Network Model

A feed-forward neural network is a type of artificial neural network in which there is no cycle in the connections between the nodes. A recurrent neural network, in which particular paths are cycled, is the reverse of a feed-forward neural network. Since the input is only processed in one direction, the feed-forward model is the simplest type of neural network. Although the data may flow via several buried nodes, it always proceeds forward and never backward. The input layer, output layer, hidden layer, neuron weights, and activation function make up feed-forward neural networks. The hardest part of creating a neural network model is obtaining precise parameters that allow for acceptable accuracy without over- or underfitting. We may experience overfitting if our model performed remarkably well on train data but poorly on test data. Given their complexity, neural networks are more likely to overfit. A data model that is under-fitted has a high error rate on both the training set and unobserved data because it is unable to effectively represent the relationship between the input and output variables. It arises when a model is overly simplistic, which might happen when a model needs more input information or training time. Regularization is a technique that modifies the learning procedure slightly such that the model generalizes more successfully. The model then performs better on the unobserved data as a result.

How to decide on the number of hidden layers and nodes in a feedforward neural network is the first query that arises. The number of neurons in the input layer is the same as the number of features in our input data. Every NN has precisely one output layer, the same as the input layer. The number of neurons in this layer may be easily calculated; it is entirely dependent on the model configuration that has been selected. Initially, the number of hidden layers is set to one and the number of neurons in the first hidden layer is set to be the mean of the neurons in the input and output layers.

The optimization of the network configuration is the following stage. By varying the hidden layer's number of neurons and batch size, we have created and validated a large variety of models using training and validation data. Batch size is the number of samples that are processed before the model is changed. Whereas epoch means the number of complete iterations through the whole training dataset. A batch must have a minimum size of one and a maximum size that is less than or equal to the number of samples in the training dataset.

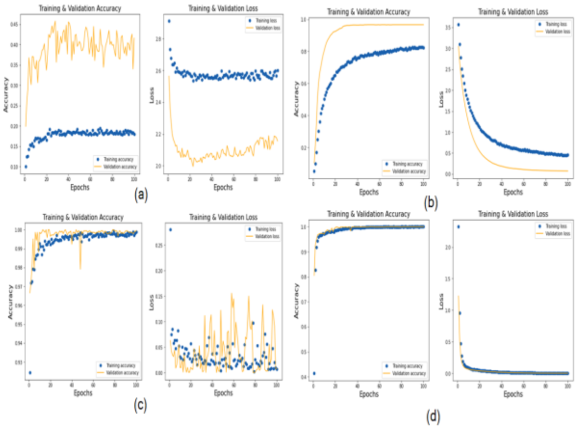

Figure 5: Fig (a) represents the training vs validation accuracy and training vs validation loss of 2 neurons in the hidden layer and batch size 2. (b) represents the training vs validation accuracy and training vs validation loss of 22

neurons in the hidden layer and batch size 512. (c) represents the training vs validation accuracy and training vs validation loss of 512 neurons in the hidden layer and batch size 2. (d) represents the training vs validation accuracy and training vs validation loss of 512 neurons in the hidden layer and batch size 512.

The figures demonstrate that learning occurs more quickly in small batches, but that the learning process is unstable and more variable in terms of classification accuracy. Larger batch sizes slow down learning, but in the end, a more stable model is reached, as seen by a smaller variance in classification accuracy.

One input layer, one hidden layer, and one output layer make up the Feed-Forward Neural Network (FFNN) architecture utilized in this study to categorize speakers, as seen in Figure 4. 193 neurons, or the number of characteristics in each speaker utterance, were used in two separate FFNN models in the input layer. There were 59 neurons in each concealed layer. Rectified Linear Unit (ReLU) was the activation function utilized in each buried layer. The output layer computed the multiclass classification output values using the softmax transfer function.

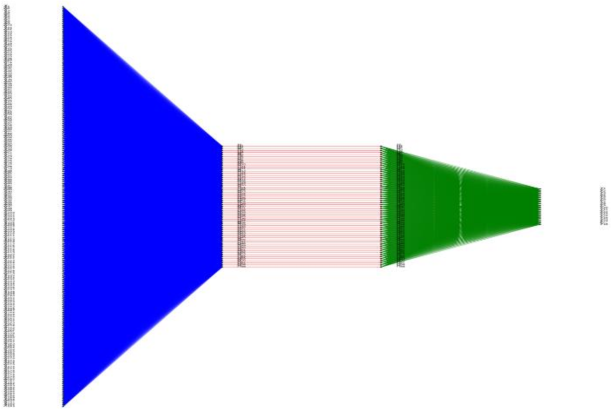

Figure 6: Feed Forward Neural Network Architecture for Speaker Identification Experiment 193 input,18 output, and 1 hidden layer with 59 neurons.

IV. EXPERIMENTAL RESULTS

This section presents all the results of the experiments performed in this study. The first experiment was to classify the speakers using a feed-forward neural network with text-dependent voice samples. The second experiment was classifying the speakers using a feed-forward neural network and with the help of text-independent voice samples. Third, we have compared the results of the previous two experiments with different classification algorithms: SVM, Decision Tree, and random forest (RF).

A. Result of Experimental Setting I

In this section, we present the result of Experiment Setting I, where the feature vector of 193 components is fed to the feed-forward neural network. Speaker Identification Accuracy, Precision, Recall, and F1-Score are used to describe the experimental outcome of the suggested work. To guarantee the consistency of the outcome, additional metrics like precision, recall, and F1 score were also assessed. Table 2 displays the outcomes of the same. Setting up the experiment as suggested produced an accuracy of 97.98.

Table 2: Accuracy, Precision, recall, and f1-score value of each Speaker of Experiment Setting I

	precision	recall	f1-score	support
SP1	0.982759	0.982759	0.982759	58
SP2	1	0.982759	0.991304	58
SP3	0.982143	0.964912	0.973451	57
SP4	0.982759	0.982759	0.982759	58

SP5	0.966102	0.982759	0.974359	58
SP6	1	0.965517	0.982456	58
SP7	0.948276	0.964912	0.956522	57
SP8	1	0.965517	0.982456	58
SP9	0.982456	0.965517	0.973913	58
SP10	0.982759	0.982759	0.982759	58
SP11	0.982759	0.982759	0.982759	58
SP12	1	0.982759	0.991304	58
SP13	0.966667	1	0.983051	58
SP14	0.966667	1	0.983051	58
SP15	0.983051	1	0.991453	58
SP16	0.949153	0.965517	0.957265	58
SP17	0.982143	0.964912	0.973451	57
SP18	0.982759	1	0.991304	57
accuracy	0.979808	0.979808	0.979808	0.979808
macro avg	0.980025	0.979784	0.979799	1040
weighted avg	0.980049	0.979808	0.979822	1040

The number of times our model accurately predicted the correct speaker was calculated using precision. The recall measures how many successful, positive labels out of all possible labels that the model was able to identify. A weighted average of recall and precision is used to determine the F1 score.

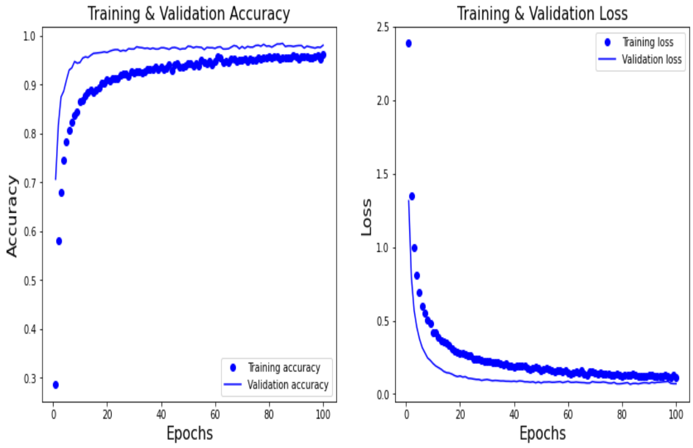

Figure 7: Training and validation accuracy and loss curve by the epoch of Experiment Setting I

From Table II, it can be deduced that for four speakers, we got a good precision and recall value of 1 and [0.96-0.98], while for the other speakers' precision range is between [0.94 – 0.98], and the recall range is between [0.96 – 0.98], which is reasonably satisfactory.

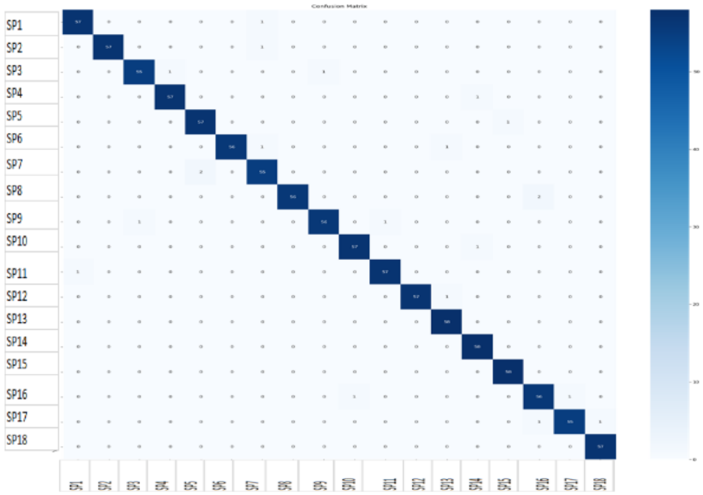

Figure 8: Confusion Matrix of Experiment I

B. *Result of Experiment Setting II*

This section represents the result of Experiment Setting II, where the feature vector of 193 components is fed to the feed-forward neural network. The Experimental result of the proposed work is also given in terms of Speaker Identification Accuracy, Precision, recall, and f1-score. The results of the same are shown in Table 3. The proposed Experiment setting gave an accuracy of 86.5 while executed.Avoid combining SI and CGS units, such as current in amperes and magnetic field in oersteds. This often leads to confusion because equations do not balance dimensionally. If you must use mixed units, clearly state the units for each quantity that you use in an equation. Training and validation accuracy and loss curve by the epoch of Experiment Setting II and Confusion Matrix of Experiment II are shown in Figure 9 and Figure 10, respectively.

Table 3: Accuracy, Precision, recall, and f1-score value of each Speaker of Experiment Setting II

	precision	**recall**	**f1-score**	**support**
SP1	0.815534	0.933333	0.870466	90.000000
SP2	0.952381	0.888889	0.919540	90.000000
SP3	0.756522	0.966667	0.848780	90.000000
SP4	0.850000	0.755556	0.800000	90.000000
SP5	0.975309	0.877778	0.923977	90.000000
SP6	0.961538	0.833333	0.892857	90.000000
SP7	0.687500	0.855556	0.762376	90.000000
SP8	0.794643	0.988889	0.881188	90.000000
SP9	0.953846	0.688889	0.800000	90.000000
SP10	0.939024	0.855556	0.895349	90.000000
SP11	0.909091	1.000000	0.952381	90.000000
SP12	0.731959	0.788889	0.759358	90.000000
SP13	0.820755	0.966667	0.887755	90.000000
SP14	0.961039	0.822222	0.886228	90.000000
SP15	0.983871	0.677778	0.802632	90.000000
SP16	0.825688	1.000000	0.904523	90.000000
SP17	0.951807	0.877778	0.913295	90.000000
SP18	0.946667	0.788889	0.860606	90.000000
accuracy	0.864815	0.864815	0.864815	0.864815
macro avg	0.878732	0.864815	0.864517	1620.000000
weighted avg	0.878732	0.864815	0.864517	1620.000000

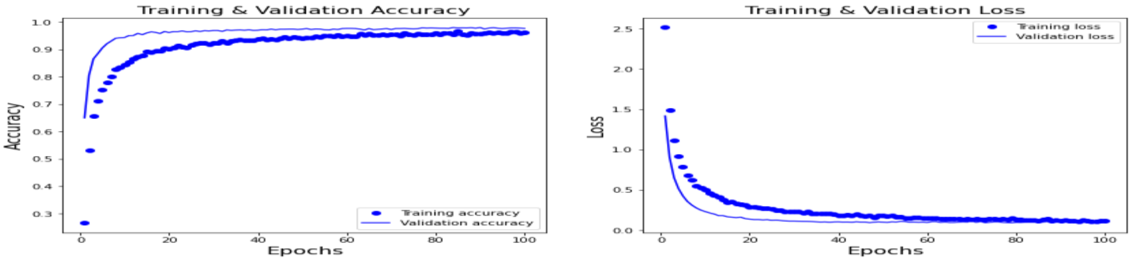

Figure 9: Training and validation accuracy and loss curve by the epoch of Experiment Setting II

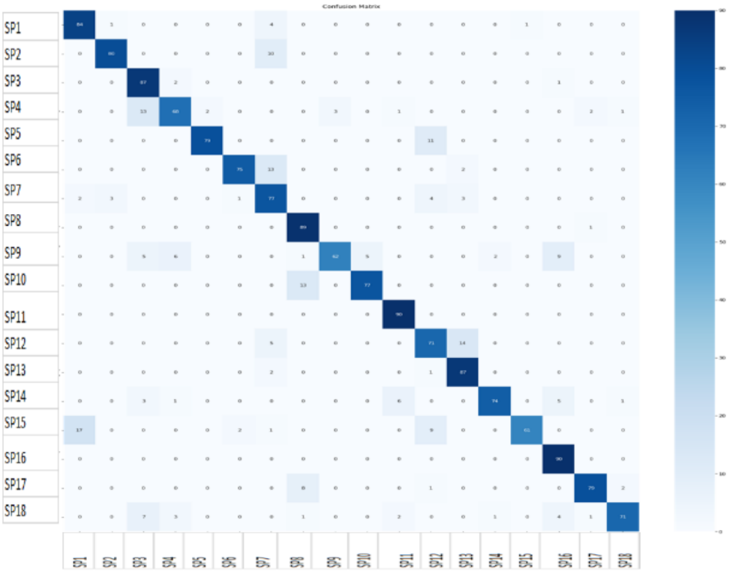

Figure 10: Confusion Matrix of Experiment II

C. *Result of Experiment Setting III*

This section represents the result of Experiment Setting III, where we have compared the results of the previous two experiments with different classification algorithms: SVM, Decision Tree, and random forest (RF). The Experimental result of the comparison work is also given in terms of Speaker Identification Accuracy. The same results are shown in Tables 4 and 5, respectively.

Random forests often outperform decision trees and support vector machines (SVMs) in terms of accuracy due to their ability to reduce overfitting and handle high-dimensional data[32] more effectively. Random forests are an ensemble learning method, meaning they combine multiple models (decision trees in this case) to improve predictive performance. This ensemble approach helps to reduce overfitting compared to individual decision trees[33].

Table 4: Comparison of SVM, Decision Tree, Random Forest, and Neural Network for the Experiment I.

Sl. No.	SVM	Decision Tree	Random Forest	Neural Network
1	83.07	76.25	96.05	97.98
2	84.50	76.80	96.50	98.00
3	82.05	76.00	95.75	97.00
4	84.00	75.60	96.00	97.75
5	84.75	76.50	95.50	96.25

Artificial neural networks (ANNs) often outperform random forests (RFs) due to their ability to capture complex nonlinear relationships in data through their layered architecture and activation functions, which is particularly advantageous for tasks involving high-dimensional data or intricate patterns. This enables them to learn intricate patterns in the data that may be beyond the capability of decision trees, which are the building blocks of random forests. A study by LeCun et al. (2015) titled "Deep Learning" [34] extensively discusses the representation power of deep neural networks, showcasing their ability to learn hierarchical representations of data.

Table 5: Comparison of SVM, Decision Tree, Random Forest, and Neural Network for the Experiment II.

Sl. No.	**SVM**	**Decision Tree**	**Random Forest**	**Neural Network**
1	67.71	58.14	82.77	86.50
2	67.00	57.88	81.45	86.00
3	68.05	58.80	83.00	87.30
4	67.20	58.08	81.80	86.30
5	67.50	57.20	82.70	87.45

D. Result of Experiment Setting IV

This section represents the result of Experiment Setting IV, where the feature vector of 194 components is fed to the feed-forward neural network. The Experimental result of the proposed work is also given in terms of Word Identification Accuracy, Precision, recall, and f1-score. The results of the same are shown in Table 6. The proposed Experiment setting gave an accuracy of 93.92 while executed.

Table 6: Accuracy, Precision, recall, and f1-score value of each Word of Experiment Setting IV

Words	precision	recall	f1-score	support
Nga001	0.952381	0.952381	0.952381	21.000000
Nga003	0.904762	0.904762	0.904762	21.000000
Nga006	0.909091	1.000000	0.952381	20.000000
Nga010	1.000000	1.000000	1.000000	21.000000
Nga011	0.904762	0.950000	0.926829	20.000000
Nga018	1.000000	0.952381	0.975610	21.000000
Nga021	1.000000	0.904762	0.950000	21.000000
Nga022	0.952381	1.000000	0.975610	20.000000
Nga025	1.000000	1.000000	1.000000	20.000000
Nga026	1.000000	1.000000	1.000000	21.000000
Ta001	0.818182	0.900000	0.857143	20.000000
Ta002	0.947368	0.900000	0.923077	20.000000
Ta003	0.869565	0.952381	0.909091	21.000000
Ta004	0.950000	0.950000	0.950000	20.000000
Ta005	1.000000	1.000000	1.000000	20.000000
Ta006	0.950000	0.950000	0.950000	20.000000
Ta007	0.850000	0.850000	0.850000	20.000000
Ta010	0.954545	1.000000	0.976744	21.000000
Ta013a	0.952381	1.000000	0.975610	20.000000
Ta015	0.850000	0.809524	0.829268	21.000000
Tha001	0.904762	0.950000	0.926829	20.000000
Tha005	0.904762	0.904762	0.904762	21.000000
Tha006	1.000000	0.750000	0.857143	20.000000
Tha010	0.952381	0.952381	0.952381	21.000000
Tha012	0.904762	0.904762	0.904762	21.000000

Tha013	1.000000	1.000000	1.000000	20.000000
Tha014	0.863636	0.950000	0.904762	20.000000
Tha016	1.000000	0.950000	0.974359	20.000000
Tha019	0.952381	1.000000	0.975610	20.000000
Tha022	1.000000	1.000000	1.000000	20.000000
Wa001	0.842105	0.800000	0.820513	20.000000
Wa003a	1.000000	1.000000	1.000000	20.000000
Wa003b	1.000000	1.000000	1.000000	20.000000
Wa004	1.000000	0.950000	0.974359	20.000000
Wa006	0.857143	0.900000	0.878049	20.000000
Wa008	0.809524	0.850000	0.829268	20.000000
Wa010	0.875000	1.000000	0.933333	21.000000
Wa011	0.894737	0.809524	0.850000	21.000000
Wa014	1.000000	0.950000	0.974359	20.000000
Wa015	0.952381	0.952381	0.952381	21.000000
Ya001	1.000000	1.000000	1.000000	21.000000
Ya001a	0.904762	0.950000	0.926829	20.000000
Ya002a	0.944444	0.850000	0.894737	20.000000
Ya003a	0.900000	0.900000	0.900000	20.000000
Ya004	1.000000	0.904762	0.950000	21.000000
Ya005	0.904762	0.950000	0.926829	20.000000
Ya008a	1.000000	0.952381	0.975610	21.000000
Ya008b	0.952381	0.952381	0.952381	21.000000
Ya008c	0.952381	1.000000	0.975610	20.000000
Ya008d	1.000000	0.952381	0.975610	21.000000
accuracy	0.939216	0.939216	0.939216	0.939216
macro avg	0.940754	0.939238	0.938980	1020.000000
weighted avg	0.940904	0.939216	0.939048	1020.000000

Training and validation accuracy and loss curve by the epoch of Experiment Setting IV and Confusion Matrix of Experiment IV are shown in Figure 11 and Figure 12, respectively.

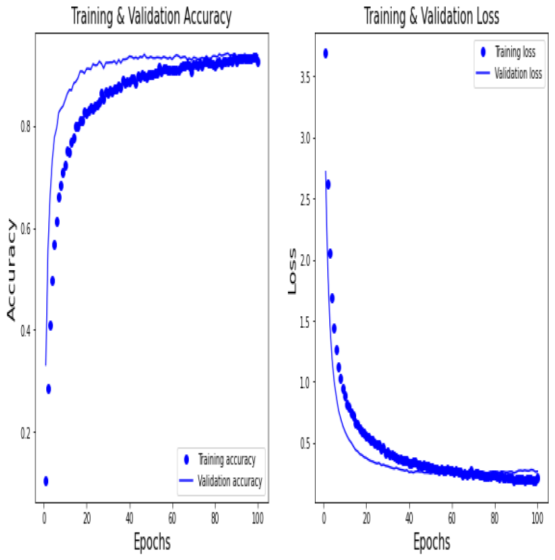

Figure 11: Training and validation accuracy and loss curve by the epoch of Experiment Setting IV

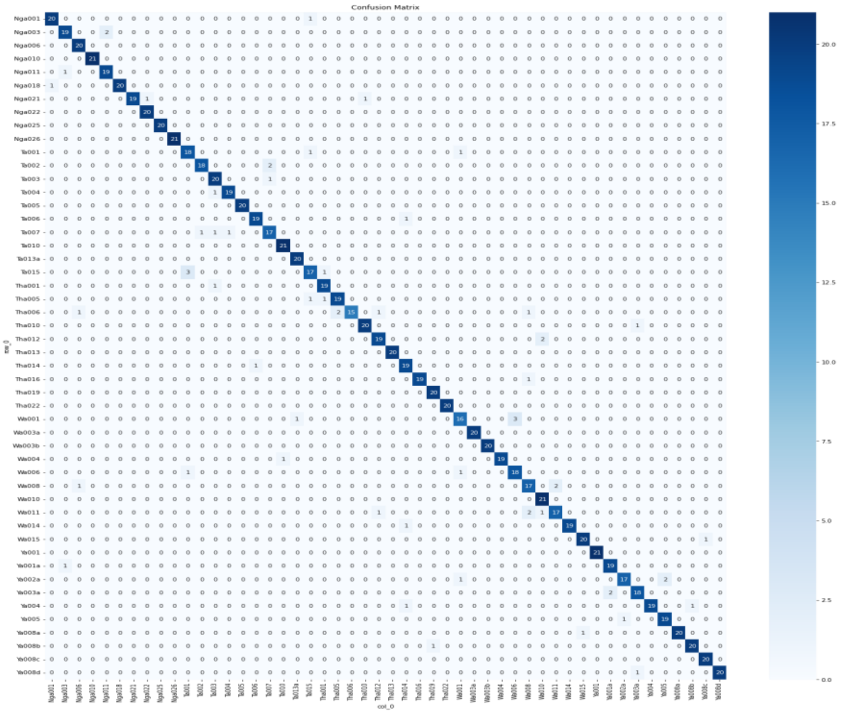

Figure 12: Confusion Matrix of Experiment IV

E. *Result of Experiment Setting V*

In this experiment, we tried to identify words irrespective of Speaker, but speaker-independent word identification for the low-resource language Tai-Phake yields poor results when using classical machine learning and neural network methods due to several key factors.

Firstly, the lack of sufficient labeled data is a fundamental challenge. Low-resource languages typically have limited available datasets for training models. This scarcity restricts the ability of machine learning algorithms to learn robust patterns and variations in speech features specific to the language, such as phonetic nuances and dialectal differences. Secondly, the linguistic characteristics of Tai-Phake, as with many low-resource languages, differ significantly from widely studied languages like English or Hindi. Neural networks and traditional machine learning models trained on datasets from these dominant languages may not generalize well to Tai-Phake due to phonetic differences, tonal variations, and unique speech patterns. Thirdly, the acoustic properties of Tai-Phake speech may not align with the assumptions made by standard models. Neural networks, for instance, rely on patterns in spectrograms or other acoustic representations of speech. If these patterns do not match well with Tai-Phake speech characteristics, the model's performance can suffer significantly.

Moreover, the problem of speaker independence exacerbates these challenges. Speaker-independent systems aim to recognize speech from any speaker, requiring models to generalize across different voices and speaking styles. In low-resource languages, where training data is limited, achieving robust speaker independence becomes particularly challenging.

In summary, the poor results observed in speaker-independent word recognition for Tai-Phake using classical machine learning and neural network methods stem from the scarcity of labeled data, linguistic and acoustic dissimilarities from dominant languages, and the difficulty in achieving speaker independence under constrained data conditions.

V. DISCUSSION

As per the experimental findings of the current work, the proposed feature set(MFCC, Chroma, Mel, Contrast, and Tonnetz) and FFNN can categorize speaker utterances with an overall accuracy ranging from 86.00% to 98.00%. The proposed feature set and FFNN demonstrated the maximum accuracy and outperformed Experiment Setting I, as seen in the experimental results (Experiment Setting II). The fact that Experiment Setting I was done with text previously spoken by the speakers, whereas Experiment Setting II was done with text-independent voice, could be one of the reasons for its subpar performance. As a result, the classifier in Experiment Setting I can categorize speaker voice signal patterns with more accuracy and lower classification error.

Section III-B discusses the results of the Experimental setup II. Three different machine learning classifiers were outperformed by the FFNN classifier. The FFNN classifier offers a superior discriminative ability for speaker identification by effectively recognizing complex and nonlinear patterns from high-dimensional datasets [24][35].

The results of this research paper demonstrate promising advancements in speaker identification, particularly for low-resource languages. Achieving an accuracy of 98% for text-dependent and 86% for text-independent speaker identification using two different neural network architectures signifies a significant breakthrough in addressing the challenges associated with identifying speakers in languages with limited available data. One of the notable aspects of this study is the utilization of neural network architectures tailored to the specific characteristics of the low-resource language. By customizing the architecture to the linguistic nuances and phonetic variations present in the target language, the models could effectively extract discriminative features for speaker identification. The achieved accuracies, though commendable, also shed light on areas for potential improvement. Despite the impressive results, there remains a performance gap, especially in the case of text-independent speaker identification. Further research is needed to explore techniques for enhancing the robustness of the models, particularly in scenarios where speech samples may vary widely in terms of content and context. Moreover, the scalability and generalizability of the proposed approach need to be investigated. While the current study focuses on a specific low-resource language, extending the methodology to other languages with similar characteristics could provide insights into the adaptability of the models across different linguistic contexts. Additionally, the impact of various factors such as speaker demographics, environmental conditions, and recording quality on the performance of the identification system warrants attention. Understanding how these factors influence the reliability and accuracy of the models is crucial for real-world deployment, especially in diverse and dynamic settings.

Investigating techniques for data augmentation could help alleviate the limitations imposed by the scarcity of training data. Augmenting the existing dataset through methods such as pitch shifting, time warping, and noise injection could potentially enhance the robustness and generalization capabilities of the models. This research can lead to the development of a mobile application for the Tai-Phake speaker recognition system.

References

[1] Tu, Y., Lin, W. and Mak, M.W., 2022. A survey on text-dependent and text-independent speaker verification. *IEEE Access*, *10*, pp.99038-99049.

[2] El-Moneim, S.A., Sedik, A., Nassar, M.A., El-Fishawy, A.S., Sharshar, A.M., Hassan, S.E., Mahmoud, A.Z., Dessouky, M.I., El-Banby, G.M., El-Samie, F.E.A. and El-Rabaie, E.S.M., 2021. Text-dependent and text-independent speaker recognition of reverberant speech based on CNN. *International Journal of Speech Technology*, *24*(4), pp.993-1006.

[3] Pisoni, David B., Howard C. Nusbaum, Paul A. Luce, and Louisa M. Slowiaczek. "Speech perception, word recognition and the structure of the lexicon." *Speech communication* 4, no. 1-3 (1985): 75-95.

[4] Diller, Anthony, Jerry Edmondson, and Yongxian Luo. *The Tai-Kadai Languages*. Routledge, 2004.

[5] Rohdin, Johan, et al. "End-to-end DNN based speaker recognition inspired by i-vector and PLDA." *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2018.

[6] Snyder, David, et al. "Deep neural network embeddings for text-independent speaker verification." *Interspeech*. Vol. 2017. 2017.

[7] Li, Ruirui, et al. "Automatic speaker recognition with limited data."

[8] Ye, Feng, and Jun Yang. "A deep neural network model for speaker identification." *Applied Sciences* 11.8 (2021): 3603.

[9] Bai, Zhongxin, and Xiao-Lei Zhang. "Speaker recognition based on deep learning: An overview." *Neural Networks* 140 (2021): 65-99.

[10] Jahangir, Rashid, et al. "Text-independent speaker identification through feature fusion and deep neural network." *IEEE Access* 8 (2020): 32187-32202.

[11] Jahangir, Rashid, et al. "Speaker identification through artificial intelligence techniques: A comprehensive review and research challenges." *Expert Systems with Applications* 171 (2021): 114591.

[12] An overview of text-independent speaker recognition: From features to supervectors." *Speech communication* 52.1 (2010): 12-40.

[13] Ayvaz, Uğur, et al. "Automatic Speaker Recognition Using Mel-Frequency Cepstral Coefficients Through Machine Learning." *Computers, Materials & Continua* 71.3 (2022).

[14] Wang, Shuai, et al. "Discriminative neural embedding learning for short-duration text-independent speaker verification." *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 27.11 (2019): 1686-1696.

[15] Maurya, A., Kumar, D., & Agarwal, R. (2018). Speaker Recognition for Hindi Speech Signal using MFCC-GMM Approach. *Procedia Computer Science 125*, 880-887.

[16] Mansour, A., &Lachiri, Z. (2017). SVM based Emotional Speaker Recognition using MFCC-SDC Features. *International Journal of Advanced Computer Science and Applications(IJACSA), 8(4)*.

[17] Chelali, F., & Djeradi, A. (2017). Text dependant speaker recognition using MFCC, LPC and DWT. *International Journal of Speech Technology. 20*, 725–740.

[18] Liu, Jung-Chun, et al. "An MFCC-based text-independent speaker identification system for access control." *Concurrency and Computation: Practice and Experience* 30.2 (2018): e4255.

[19] Nasr, M.A., Abd-Elnaby, M., El-Fishawy, A.S. *et al.* (2018) Speaker identification based on normalized pitch frequency and Mel Frequency Cepstral Coefficients. *International Journal of Speech Technology* **21,** 941–951.

[20] Dutta, Munmi, et al. "Closed-set text-independent speaker identification system using multiple ANN classifiers." *Proceedings of the 3rd International Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA) 2014*. Springer, Cham, 2015.

[21] Jahangir, Rashid, et al. "Text-independent speaker identification through feature fusion and deep neural network." *IEEE Access* 8 (2020): 32187-32202.

[22] Bharti, R., & Bansal, P. (2015). Real Time Speaker Recognition System using MFCC and Vector Quantization Technique. *International Journal of Computer Applications 117(1)*, 25-31

[23] Y. Lei, N. Scheffer, L. Ferrer, and M. McLaren, "A novel scheme for speaker recognition using a phonetically-aware deep neural network," in ICASSP, 2014.

[24] Y. Lei, L. Ferrer, M. McLaren, and N. Scheffer, "Comparative study on the use of senone-based deep neural networks for speaker recognition," Submitted to IEEE Trans. ASLP, 2014.

[25] Garcia-Romero D., Zhang X., McCree A., and Povey D., "Improving speaker recognition performance in the domain adaptation challenge using deep neural networks," in SLT, 2014.

[26] P. Matějka et al., "Analysis of DNN approaches to speaker identification," 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2016, pp. 5100-5104, doi: 10.1109/ICASSP.2016.7472649.

[27] Dimitri Palaz, Mathew Magimai.-Doss, and Ronan Collobert, "Analysis of CNN-based speech recognition system using raw speech as input," in Proc. of Interspeech, 2015

[28] H. Muckenhirn, M. Magimai.-Doss and S. Marcell, "Towards Directly Modeling Raw Speech Signal for Speaker Verification Using CNNS," 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2018, pp. 4884-4888, doi: 10.1109/ICASSP.2018.8462165.

[29] A. M. Jalil, F. S. Hasan and H. A. Alabbasi, "Speaker identification using convolutional neural network for clean and noisy speech samples," 2019 First International Conference of Computer and Applied Sciences (CAS), 2019, pp. 57-62, doi: 10.1109/CAS47993.2019.9075461.

[30] M. M. Hossain, B. Ahmed and M. Asrafi, "A real time speaker identification using artificial neural network," 2007 10th international conference on computer and information technology, 2007, pp. 1-5, doi: 10.1109/ICCITECHN.2007.4579414.

[31] Pawar, Rupali &Kajave, P. & Mali, Suresh. (2005). Speaker Identification using Neural Networks.. 429-433.

[32] Liaw, Andy, and Matthew Wiener. "Classification and regression by randomForest." R news 2.3 (2002): 18-22.

[33] Breiman, Leo. "Random forests." Machine learning 45 (2001): 5-32.

[34] LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. "Deep learning." *nature* 521.7553 (2015): 436-444.