Design and Development of Voice OTP Authentication System

Pavanitha Manche(✉), Sahaja Nandyala, Jagabandhu Mishra, Gayathri Ananthanarayanan, and S. R. Mahadeva Prasanna

Indian Institute of Technology Dharwad, Dharwad 580011, India
{200010027,200010032,jagabandhu.mishra.18,gayathri,prasanna}@iitdh.ac.in

Abstract. Voice OTP Authentication (VOA) provides authorization for a speaker by validating the spoken One-Time Password(OTP) and the speaker's identity. Even though Speaker Recognition and Digit Recognition techniques are fairly mature, the exploration in the direction of the development of VOA systems is limited. This work proposes a speaker and speech representation-based framework to develop the VOA system. Our design uses ECAPA-TDNN based speaker representation and wav2vec conformer-based digit representation to perform VOA. The achieved performance of the speaker identification, OTP identification, and the combined VOA system in the DigitUtter-IITDH dataset in terms of identification accuracy are 96.75%, 83.25% and 78.92%, respectively. Further, to deploy the VOA system on an edge device, we conduct a comprehensive performance analysis by deploying the proposed VOA system from a high-end server class machine to an embedded edge device. Our experimental results indicate that the average inference time for an OTP Authentication using an edge device is 3.14 seconds, while it takes 0.05 seconds on the server class system.

Keywords: Voice OTP · Speaker Identification · Digit Recognition

1 Introduction

The proliferation of businesses across the world with remote work and distributed teams necessitates the development and deployment of remote authentication systems [3,13]. Remote authentication systems allow users to access their accounts and resources from anywhere with an active internet connection. ***Voice OTP Authentication*** **(VOA)** is one such system that provides two-level authentication: (1) verifies the speaker's identity and (2) verifies the spoken OTP. `Speaker Recognition (SR)` and `Digit Recognition (DR)` are matured speech technologies and provide acceptable performance in practical deployments [3,17]. It also leverages the uniqueness of an individual's voice, making it difficult for unauthorized users to impersonate someone else. It is particularly beneficial for users with disabilities or those with difficulty typing or using traditional authentication methods. Hence, compared to other authentication alternatives, the VOA system is preferable as a low-cost, easy-to-use

A. Karpov et al. (Eds.): SPECOM 2023, LNAI 14339, pp. 513–528, 2023.
https://doi.org/10.1007/978-3-031-48312-7_41

solution. The VOA system can have a broad spectrum of applications and is not limited to online banking, remote logins to workstations, attendance systems, etc. However, the work related to VOA is still at a nascent stage in the literature.

In this work, we propose a VOA system framework, and Fig. 1 depicts a practical use case of the same. In the proposed framework, the user requests the authentication system to access his personal data. The authentication system in turn, sends a request to the `OTP generator module` to generate a N digit OTP, and the generated OTP is forwarded to the user. The user utters the received OTP, and using the user's utterance, the authentication system verifies the speaker's identity and deciphers the spoken OTP. Finally, if the identity and spoken OTP are correct, the authentication system provides access to the personal data and asks the user for another attempt if otherwise.

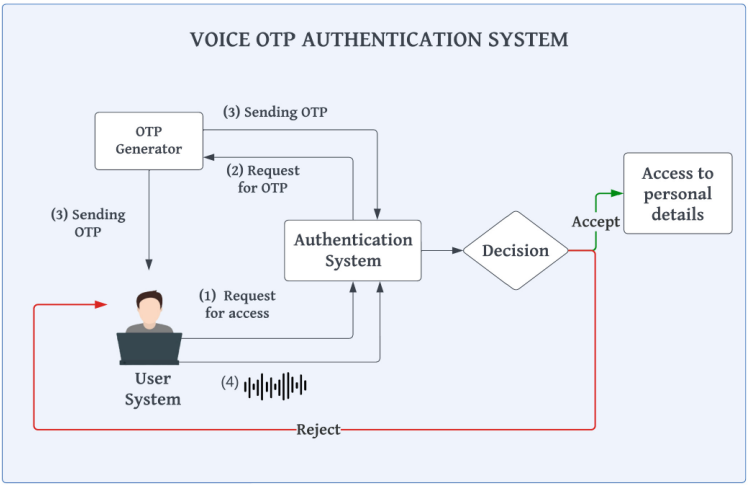

Fig. 1. Working of Voice OTP Authentication system.

In the early days, SR and DR were explored by proposing various feature extraction and modelling techniques [5,13]. The feature-based techniques have evolved over time by analyzing the production and perception mechanism of the speech signal [11,13–15,18]. The `Mel frequency cepstral coefficient (MFCC)`, and `perceptual linear prediction (PLP)` have been proposed with respect to the perception mechanism of speech [12,13] while the `linear prediction cepstral coefficient (LPCC)`, `residual LPC`, and `residual phase-based` features have been proposed for the production mechanism of speech [10,16,18]. Mostly, the `formant (spectral resonances) locations and their dynamics` play an important role in identifying both speaker and digit, hence parameterized in different ways [4,13]. Out of them, `MFCC` has shown to be successful over the rest for both digit and Speaker Recognition

tasks [4,13]. Further, the features are modelled using `vector quantization (VQ)`, `Gaussian mixture model (GMM)`, and `i-vector` to perform the SI task while `GMM-hidden Markov model (GMM-HMM)` is largely used [5,8,13] for digit recognition. As acoustic features are sensitive to changes in acoustics in terms of device and environment variation, these systems are always built in a controlled scenario by constraining on the particular type of recording device and environment.

Deep Learning has become ubiquitous with applications across various fields. Recently, various Speaker Recognition techniques have been proposed using deep learning frameworks. Some of the existing works in the literature are Deep Neural Network–Hidden Markov Model (DNN-HMM) [17], Deepspeech2 [1], wav2vec-transformer [2], and wav2vec-conformer models [7] used to perform speech recognition task. Similarly, starting from the DNN-i-vector system, d-vector, x-vector, and emphasized channel attention, propagation, and aggregation (ECAPA-TDNN) based x-vector approaches have been proposed to perform the speaker identification task. The limitation of the traditional system design to a particular recording device and environment is relaxed by the use of pre-trained open-sourced task-specific deep learning models [2,3]. These models are trained with large amounts of speech data to perform a particular task. Further, these models achieve better performance for the SI task even when trained with a small utterance duration. However, the DI task in a given language is almost zero-shot. Motivated by these assumptions, the hypothesis is that *the use of speaker representation from the ECAPA-TDNN model* (trained in VoxCeleb [9]) and *digit representation from conformer-based wav2vec (W2V) model* (fine-tuned with Indian English [6]) can be helpful in developing the VOA system.

This work initially performs speaker identification (SI) with traditional approaches by considering MFCC-VQ and MFCC-GMM frameworks and compares the performance with the ECAPA TDNN-based speaker representation framework. Further, this work proposes a framework for training the speaker representation by generating utterances corresponding to the fixed OTP sequences. We then use the conformer-based W2V model to decode the uttered OTP sequence. We also augment this decoder with a rule-based wrapper algorithm to improve the accuracy of the decoded output. We use the combination of the output of the decoded OTP and the output of SI to evaluate the performance of the VOA system. Furthermore, for studying the feasibility of deploying the VOA system with different end devices, we consider the inference time as well as the run-time memory requirements.

The rest of the paper is organized as follows: Sect. 2 discusses the motivation for using the speaker and digit representation for designing the VOA system. Section 3 details the database used in this work. Section 4 provides the details of the proposed framework for the VOA system, while Sect. 5 discusses the experimental results. Finally, Sect. 6 presents the summary and future work directions.

2 Motivation for Using the Speaker and Digit Representations

We begin with the t-Distributed Stochastic Neighbor Embedding (t-SNE) distribution of the speaker and digit representations. The aim of this is to see whether the speaker representations of each speaker form a different cluster. Similarly, whether the digit representations of each digit form a different cluster. Figure 2 provides the t-SNE plots of speakers and digits representations.

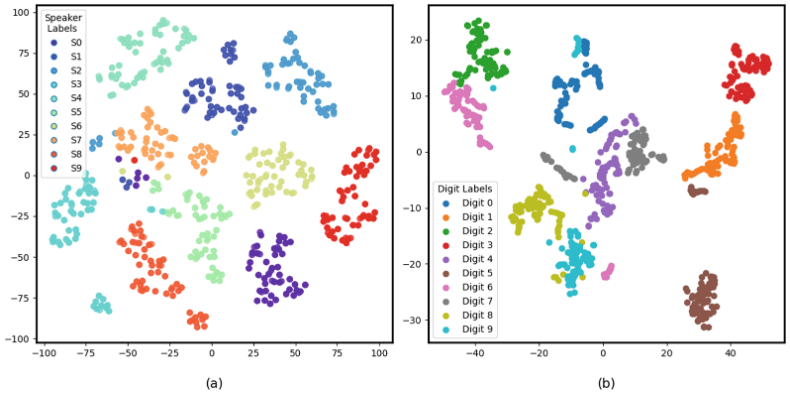

Fig. 2. t-SNE Visualization of (a) Speaker representations and (b) digits representations.

For this t-SNE study, we use utterances from 10 speakers and 100 utterances from each speaker. We consider any two random digits uttered by the speaker to observe the speaker's discrimination. Similarly, for each digit, we consider 10 utterances spoken by 10, different speakers to observe the digit discrimination. To obtain the speaker's representation for a given utterance, the filter bank features are extracted from the speech signal by considering 25 msec as the frame size and 10 msec as the frameshift. The filter bank features are then passed through the ECAPA-TDNN model (trained in the Voxceleb data and available at[1].) to obtain the 192 dimensional speaker representations. The obtained speaker representations for all the 10 speakers are projected in 2 dimension using t-SNE. The two-dimensional vectors are depicted in Fig. 2(a). From the figure, we can observe that the speaker representations are forming clusters with respect to the speakers, and overlapping between them is significantly less. This motivates us to use the ECAPA-TDNN-based speaker representations to perform the SI task. Similarly, as the W2V model works on signal level, the speech utterances are directly passed through the finetuned model (fine-tuned with 700 hours of Indian English data and available at[2]) to obtain the speech representations in every 20

[1] https://catalog.ngc.nvidia.com/orgs/nvidia/teams/nemo/models/ecapa_tdnn.
[2] https://github.com/Open-Speech-EkStep/vakyansh-models.

msec. For a given digit utterance, the speech representations are statistically pooled to obtain the digit representation. The digit representations for each digit are also projected using t-SNE to a two-dimensional plane, and the same is depicted in Fig. 2(b). From the figure, we can observe that similar to speakers', the digit representations also form distinct clusters, thus motivating us to use the digit representations to perform the OTP identification task.

3 Details of the Datasets

In this work, we use two databases: (1) In-house data and (2) Kaggle MNIST digit data to develop the VOA system. The dataset details are provided in the following subsections and summarized in Table 1.

3.1 In-House Dataset

The dataset was collected from students at IIT Dharwad, consisting of contributions from 50 speakers, with 47 being male and 3 being female speakers. The average age of the speakers is 20. Each speaker is asked to give 4 sessions, and in each session, speakers are asked to utter the digits $0-9$ with a pause after each digit. Further, after listening to the utterance, the speech belonging to each digit is manually segregated. The collected data is referred to as `DigitUtter-IITDH` dataset. The collected dataset is available at[3]DigitUtter-IITDH-dataset.

3.2 Kaggle MNIST Digit Dataset

The Kaggle MNIST Digit dataset is used as reference data to perform the initial experiments[4]. This dataset comprises voice recordings of 60 speakers, out of which 48 are male and 12 are female, all with an American accent. Each speaker contributed 50 sessions dedicated to pronouncing the digits 0 to 9. Throughout this work, we refer to this dataset as the `Kaggle Data`.

Table 1. Summary of Datasets.

Dataset	In-house Data	Kaggle Data
Speakers	50	60
Sessions	4	50
Digits	0–9	0–9

[3] https://github.com/mcqueen444/DigitUtter-IITDH-dataset.
[4] https://github.com/soerenab/AudioMNIST.

4 Proposed Framework for the Voice OTP Authentication System

The proposed VOA system consists of four modules `(1) OTP Generation`, `(2) Speaker Identification`, `(3) OTP Identification`, and `(4) Decision Logic`. Figure 3 presents the block diagram of the proposed VOA system. The details of each module are provided in the following subsections.

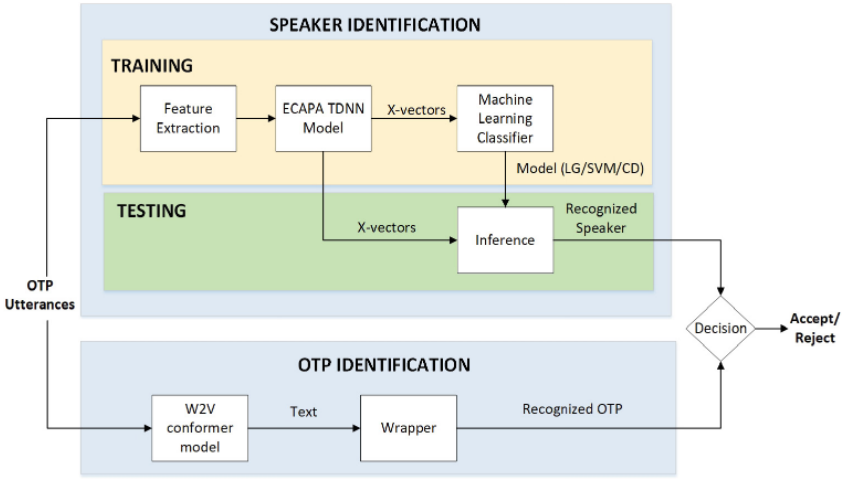

Fig. 3. Block diagram for Voice OTP Authentication system. LR, SVM, and CD denote Logistic Regression, Support Vector Machine, and Cosine Distance, respectively.

4.1 OTP Generation

In this work, OTPs with varying lengths of $1-4$ digits were generated using both DigitUtter-IITDH and Kaggle datasets. Using the DigitUtter-IITDH dataset, for a fixed OTP length of N digits, all possible combinations (i.e. 10^N) are generated as different OTPs. After that, the segregated digit-specific utterances are stitched together to form OTP utterances. Figure 4 depicts the OTP utterance generation process. The OTP utterances belonging to the first three sessions are used for training, and the fourth session is used for testing. The number of generated OTP utterances per speaker per session and the train test split is summarized in Table 2. Further, while generating OTP utterances from the Kaggle data, to make the number of training and testing utterances the same as the DigitUtter-IITDH data, the digits are sampled randomly from the sessions $1-40$ for training and $41-50$ for testing.

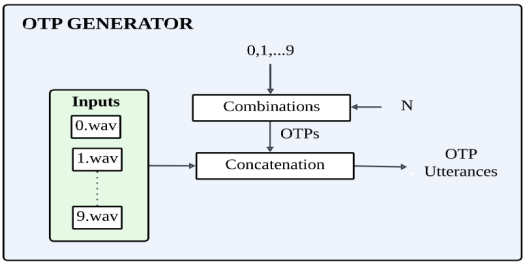

Fig. 4. Working principle of OTP Generator.

Table 2. Summary of OTP datasets that are generated, N refers to the Number of Digits in OTP.

Number of digits in the OTP (N)	1	2	3	4
Utterances per Speaker per Session	10	100	1000	10000
Total Training Utterances per Speaker	30	300	3000	30000
Total Testing Utterances per Speaker	10	100	1000	10000

4.2 Speaker Identification

After generating the OTP utterances, the speaker representations are extracted from the ECAPA-TDNN model. The ECAPA-TDNN model takes the filter bank features extracted from the speech signal (by considering 0.025 as the frame size and 0.01 secs as the frameshift) as input and is trained to classify the speakers. The model has several TDNN layers, a temporal pooling layer, and some fully connected layers. The architecture details can be found in [9]. The TDNN layers work on the frame level, the temporal pooling layer pools the frame-level information of a given utterance to a fixed-dimension vector, and then the fully connected layers work on the utterance level to classify the speaker. After training, the classifier layer is detached from the network and is used as a speaker representation extractor.

In this study, we use the ECAPA-TDNN model available in the NVIDIA NeMo toolkit [5]. The ECAPA-TDNN model is already pre-trained using the development set of the VoxCeleb-1 and two datasets having 7205 speakers with several thousands of hours of speech data. After obtaining the filter bank features from each utterance, the filter bank features are used to extract the speaker representations from the ECAPA-TDNN model. The extracted speaker representations from the ECAPA-TDNN model are well known as `x-vectors` [9]. We leverage the extracted x-vectors in two ways: (1) model-based and (2) model-free, to perform the SI task. The model-based approach includes the training of classifiers like Logistic Regression (LR) and Support Vector Machine (SVM), whereas the model-free approach uses a simple Cosine Distance (CD) based comparison.

[5] https://catalog.ngc.nvidia.com/orgs/nvidia/teams/nemo/models/ecapa_tdnn.

4.3 OTP Identification

The OTP Identification needs to be performed by decoding the OTP utterances through an Automatic Speech Recognition(ASR) model trained in Indian English. We use the W2V-based conformer model, trained and open-sourced by the Vakyansh team [6] to perform the OTP decoding. The W2V model training is generally done in two stages: (1) pre-training and (2) fine-tuning. The model is pre-trained using unlabeled speech data from 39 Indian languages of approximately 35000 hours. The pre-trained network is then finetuned with 700 hours of labelled Indian English speech data. The fine-tuned model is available at vakyansh-models.

This work uses the fine-tuned ASR model to decode the OTP utterances in the testing phase of the framework. The decoder generally outputs the orthographic form of the numbers. Hence, a wrapper algorithm is designed to convert them to numeric output. Further, in some cases, it is observed that instead of "two", "to" is decoded. The wrapper algorithm has also handled the same issue, i.e., if any of the letters are missing in the decoding, the same is substituted and converted to the corresponding numerical value. The wrapper algorithm is explained in Algorithm 1.

The decoded OTP from the W2V-based conformer model is directly given as input to the wrapper. Initially, the algorithm creates a dictionary D, storing all possible sub-words for each digit. Each sub-word is paired with its corresponding digit as the value. To construct D, the algorithm takes the alphabetical representation of the digits 0–9 in a list T. It then iterates over each element in T to generate all possible sub-words for each number, starting from a minimum length of 2 letters. Sub-words with only one letter are not considered as they cannot uniquely represent a digit. Once D is prepared, the decoded OTP words are processed individually. The algorithm checks each word against the keys in D to convert each digit to its original form. This process allows the algorithm to decode the OTP successfully.

4.4 Decision Logic

In this module, the final decision is made for the VOA system by combining the outputs given by the SI and the OTP Identification module. The identified speaker and decoded OTP are verified with the claimant's identity and generated OTP. The system will accept the trial only if both decisions are positive. If any one of the decisions is negative, the trial will be rejected.

5 Experimental Results and Discussions

In this section, we discuss the performance of the proposed system, along with the time to identify the speaker as well as the decoding of the OTP (`inference time`) and `runtime memory consumption` on various end devices.

We conducted several experiments to explore various aspects of the proposed VOA system.

Algorithm 1: Digit Recognition Wrapper

```
Input  : transcription
Output: pred
T ← ["zero", "one", "two", "three", "four", "five", "six", "seven", "eight", "nine"];
D ← {} ;                                   // dictionary for Substrings and digits
for i ← 0 to len(T) do
    S ← T[i] ;                                                    // Current text
    for L ← 2 to len(S) + 1 do
        for start ← 0 to len(S) - L + 1 do
            sub ← S[start : start + L] ;                            // Substring
            D[sub] ← i ;                                               // Store
        end
    end
end
pred ← '' ;                                                  // Prediction string
for w ∈ transcription.split() do
    if w ∈ D then
        d ← D[w] ;                                              // Retrieve digit
        pred += str(d) ;                                          // Append digit
    end
end
```

1. We begin with experiments to quantify the gains of using x-vector representations over the traditional VQ and GMM-based framework to perform the SI task.
2. We then perform extensive studies using the model-based approach by varying the OTP length N and the number of training samples to understand its impact on the SI performance
3. We then evaluate the performance of the VOA system by implementing it in different end devices.

In this study, the following three different devices with varying performance capabilities were selected to understand the performance variation and assess the feasibility of practical deployment:

- **Server**: `24-Core Intel(R) Xeon(R) W-2265 CPU @ 3.50 GHz`.
- **Desktop**: `8-Core Intel(R) Core(TM) i7-8550U CPU @ 1.80 GHz`.
- **Edge Device**: `4-core ARM Cortex-A73 CPU @ 1.80 GHz and a 2-core ARM Cortex-A53 CPU @ 1.90 GHz`.

5.1 SI with X-vector-based Speaker Representation

The aim of the experiment is to showcase the significance of the x-vector-based framework over the traditional VQ and GMM framework and quantify the achievable performance gains. We use the in-house `DigitUtter-IITDH` dataset to perform this set of experiments. From the dataset, we use the utterances from the first 3 sessions to train the VQ and GMM classifier by extracting the MFCC features from the speech signal (0.02 and 0.01 are the framesize and frameshift, respectively). We swift through a range of cluster sizes by varying the number of

clusters from 32 to 256 and found that the cluster size of 64 performs the best. Thus, in our experiments, the VQ and GMM are trained with a cluster size of 64. Similarly, we extract the x-vectors from the training utterances and then model them using SVM and LR. We use the utterances belonging to the 4^{th} session for the testing. The obtained result in terms of identification accuracy is tabulated in Table 3. From the table, it can be observed that the best performance obtained using the x-vector framework is 100%, in contrast to the best performance obtained in the traditional framework is 88%. This shows the significance of the x-vector-based framework over traditional frameworks to perform the SI task.

Table 3. Accuracy variation across various feature vectors.

Vectors	Model	Accuracy
MFCC	VQ	80%
	GMM	88%
x-vector	SVM	**100**%
	LR	98%

5.2 SI by Varying the OTP Length in both Training and Testing

This experiment aims to observe the variation in SI performance when varying OTP length during training and testing. For this, we use the OTP utterances generated from the first three sessions with a given OTP length N to extract the x-vectors and then train the LR and SVM classifier. The OTP length varies from `one` to `four`. We tabulate the obtained results with the LR classifier in Table 4 and with the SVM classifier in Table 5. From the tables, it can be observed that irrespective of the classifier, the performance improves with an increase in the OTP length. Further, it is also observed that the performance of the same OTP length training and testing is comparatively better than the cross-OTP length scenarios. We thus recommend using the same OTP length for both training and testing.

Table 4. Logistic Regression (%)

Train \ Test	1	2	3	4
1	**66.26**	53.06	42.26	23.33
2	55.28	**87.29**	87.29	81.30
3	39.72	78.05	**93.57**	92.96
4	29.54	66.13	86.58	**96.75**

Table 5. SVM (%)

Train \ Test	1	2	3	4
1	**29.34**	27.67	31.39	22.74
2	56.29	**86.09**	83.07	67.15
3	41.92	79.70	**93.99**	93.94
4	27.94	65.33	85.71	**96.40**

5.3 SI by Varying the Number of Training OTP Utterances

The aim of this experiment is to decide on the number of OTP utterances per speaker required to perform the SI task. The possible OTP utterances increase exponentially with an increase in the OTP length. Considering all the possible combinations will increase the enrollment time. Hence, we perform the SI experiment for $N = 4$ by randomly selecting 3000, 6000 OTP utterances per speaker from all possible OTP utterances and compared its performance with that of using all possible OTP utterances (i.e. 30000 per speaker). We provide the obtained performance in Table 6. Our results indicate that the performance achieved by randomly considering 6000 OTP utterance per speaker from all the possible 30000 OTP utterances is similar to the maximum achievable performance.

Table 6. Impact of Training Data Size on the SI performance.

# Utterances per Speaker in Train Data	LR Accuracy
3000	95.90%
6000	**96.75**%
30000	**96.75**%

5.4 SI with Model-Based and Model-Free Approach

We implement the SI task with model-based LR, SVM classifier, and model-free CD approach using both In-house and Kaggle data. In the CD-based approach, we extract the speaker-specific mean vectors and store them as a speaker representation. During testing, the test x-vector is compared with all the speaker representations using CD, and the speaker with the maximum cosine distance obtained is declared as the identified speaker.

We provide the obtained performance in Table 7. From the table, it can be observed that the LR with $N = 4$ performs better compared to the rest in both in-house and Kaggle datasets. The best performance in the in-house data is 96.75%, and with Kaggale data is 99.9%. The performance gap is due to the differences in the speaker's accent. Kaggle and Voxceleb have similar accents, whereas, in In-house data, all the speakers have Indian accents. Further, when using the model-free approach, the difference is even greater. The best performance achieved in In-house data is 74.45%, while in Kaggle data, it is 99.91%. Hence in the future, to further improve the performance of SI, the network should be fine-tuned with the in-house training data.

Table 7. Performance of SI

Model Name	N	In-house Data Accuracy	Kaggle Data Accuracy
SVM	1	2.00%	90.17%
	2	80.64%	99.55%
	3	94.01%	99.98%
	4	95.91%	**99.996**%
LR	1	66.27%	92.17%
	2	87.29%	99.68%
	3	94.23%	99.97%
	4	**96.75%**	**99.99**%
CD	1	53.87 %	91.83%
	2	64.68%	99.06%
	3	67.15%	99.72 %
	4	**74.45**%	**99.91**%

Table 8. Performance of OI

Number of Digits	Number of samples	Accuracy
1	500	50.70%
2	5000	74.58%
3	5000	78.89%
4	5000	**81.16**%

5.5 OTP Identification

We use the In-house data to evaluate the performance of the OTP Identification (OI) task. The total OTP utterances available with N = 1,2, 3 and 4 is 500, 5000, 50000 and 500000 (50×10^N), respectively. For evaluating the performance, instead of considering all the OTP utterances, for $N = 1$, all 500 and for $N = 2, 3$ and 4 randomly picked 5000 utterances are considered. We tabulate the resulting performance in Table 8. The OI task provides the best performance of 81.16% in terms of identification accuracy for $N = 4$. Like SI, it is observed that, with an increase in OTP length, the performance of the OI system also increases.

5.6 Performance Evaluation of VOA System

We evaluate the performance of SI and OI jointly with In-house `DigitUtter-IITDH` dataset, calculating combined accuracy by intersecting their probabilities. For SI, LR (model-based) and CD (model-free), we use the W2V conformer model and wrapper algorithm to perform the OI task. The results, as shown in

Table 9, reveal that the best accuracy achieved is 78.92% for model-based SI and 73.98% for model-free SI. The lower combined accuracy is due to the requirement for both SI and OI tasks to be simultaneously correct for authentication. Further, it can be observed from the table that the OI performance is inferior to the performance achieved in the SI task. In the future, the performance may be improved by finetuning the W2V conformer architecture with the training OTP utterances.

Table 9. The accuracy of the integrated processes involving Speaker identification (SI) and OTP Identification (OI). Note: OI is evaluated independently of LR and CD.

Model	Number of digits in test Data	SI Accuracy	OI Accuracy	Combined Accuracy
LR	1	66.26%	50.70%	32.87%
	2	87.29%	74.58%	66.14%
	3	93.58%	80.18%	75.89%
	4	96.75%	83.25%	**78.92**%
CD	1	53.09%	50.70%	33.27%
	2	64.69%	74.58%	61.10%
	3	67.15%	80.18%	65.99%
	4	74.45%	83.25%	**73.98**%

5.7 Memory Consumption Analysis

With the primary objective of implementing a VOA system on a device with limited resources, our focus was on reducing both computation time and memory usage. To address memory consumption concerns, we performed a detailed analysis of the runtime memory consumption of different steps in both tasks (SI and OI).

We conduct a comprehensive analysis of memory consumption throughout the VOA process. The modules that use larger amounts of memory are the module importation, ECAPA TDNN model loading, extraction of embeddings and vakyansh model loading (refer to Table 10). It is important to note that the memory consumption results mentioned above are specific to our model, which was trained and tested using data consisting of four-digit utterances. During the VOA process, the actual prediction stage requires less than 3 MiB of memory.

Table 10. Memory consumption at each step of the speaker verification process, CM, MI are Cumulative memory, Memory Increment at that step respectively.

Step	CM	MI
Importing modules	520 – 525 MiB	520 – 525 MiB
Loading ECAPA TDNN model	1930 – 1940 MiB	1407 – 1417 MiB
Extracting embeddings	3626 – 3628 MiB	1695 – 1698 MiB
Loading trained model and predicting	3626 – 3628 MiB	0.2 – 0.4 MiB
Loading vakyansh	4338 – 4340 MiB	711 – 713 MiB
Transcription	4340 – 4342 MiB	2.6 – 2.9 MiB

5.8 Feasibility Exploration on Different Platforms

The main objective is to study the feasibility of implementing a VOA system on low-resource devices. For this, a series of evaluations are performed to measure the proposed model's computational performance and memory usage on various devices mentioned in Sect. 5.

Table 11. Performance and Computational Time Comparison on Devices Server-CPU (SC), Laptop-CPU (LC), Odroid-N2 (OD) for Speaker Identification and OTP Identification with Logistic Regression (LR) and Cosine Distance (CD) methods.

# Digits	LR						CD					
	Accuracy(%)			Time(s)			Accuracy(%)			Time(s)		
	SC	LC	OD	SC	LC	OD	SC	LC	OD	SC	LC	OD
1	32.9	32.9	32.9	**0.0519**	0.5468	1.7357	33.3	33.3	33.3	0.0398	0.5284	1.5575
2	71.6	71.8	71.8	0.0555	0.7229	2.262	61.2	61.2	61.2	0.0424	0.9451	2.2203
3	85.4	85.4	85.4	0.0609	1.1859	2.759	63.4	63.4	63.4	0.4512	1.2341	2.9556
4	79	79	79	0.0593	1.3685	**3.1377**	70.6	70.6	70.6	0.0486	1.4675	3.0473

Table 11 presents the overall performance and computational time for both SI and OI tasks across various devices. The time values mentioned here are the average computational time taken by a particular model across the given OTPs for a specific digit. Also, the results shown here are from analyzing only a small number of OTP utterances. As seen, the accuracy scores for each device are consistent with one another. This suggests both the SI and OI modules deliver similar levels of accuracy across all the tested devices. Thus, this ensures the model can be deployed on any of the devices.

As seen from table, the time to process a 4-digit OTP on the edge device is ≈ 3 s while on the server, it is 0.05 seconds and 1.4 seconds on the desktop. Even though the edge device performs $\approx$ 50x slower than the server, with respect to the device footprint, power requirement, and cost, the edge devices are preferable

for practical deployment. In the future, the aim is to optimize the model such that the computation time in the edge device can be improved.

6 Conclusion and Future Work

In summary, this work demonstrated the significance of speaker and digit representations obtained from the ECAPA-TDNN and W2V-conformer model to develop the VOA system. The SI component provides acceptable performance, while the OI component provides a little bit inferior performance. To address this limitation in the future, the models can be further fine-tuned with the In-house training data. Further, it is also observed that except for a lag in inference time, the performance of the VOA system is stable irrespective of the devices. In the future, the aim is to optimize the VOA system for resource-constrained environments and have a plan to integrate it into real-time applications like mobile banking.

Acknowledgements. The authors would like to acknowledge the Ministry of Electronics and Information Technology (MeitY), Govt. of India, for supporting us through different projects. Additionally, the authors also acknowledge the effort of the undergraduate students, who have contributed to the development of the DigitUtter-IITDh dataset.

References

1. Amodei, D., et al.: Deep speech 2: end-to-end speech recognition in English and mandarin. In: International Conference on Machine Learning, pp. 173–182. PMLR (2016)
2. Baevski, A., Zhou, Y., Mohamed, A., Auli, M.: wav2vec 2.0: a framework for self-supervised learning of speech representations. Adv. Neural. Inf. Process. Syst. **33**, 12449–12460 (2020)
3. Bai, Z., Zhang, X.L.: Speaker recognition based on deep learning: an overview. Neural Netw. **140**, 65–99 (2021)
4. Benesty, J., Sondhi, M.M., Huang, Y.A. (eds.): Springer Handbook of Speech Processing. SH, Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-49127-9
5. Benzeghiba, M., De Mori, R., Deroo, O., Dupont, S., Erbes, T., Jouvet, D., Fissore, L., Laface, P., Mertins, A., Ris, C., et al.: Automatic speech recognition and speech variability: a review. Speech Commun. **49**(10–11), 763–786 (2007)
6. Chadha, H.S., et al.: Vakyansh: ASR toolkit for low resource Indic languages (2022)
7. Chung, Y.A., Zhang, Y., Han, W., Chiu, C.C., Qin, J., Pang, R., Wu, Y.: W2v-BERT: combining contrastive learning and masked language modeling for self-supervised speech pre-training. In: 2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU), pp. 244–250. IEEE (2021)
8. Dehak, N., Kenny, P.J., Dehak, R., Dumouchel, P., Ouellet, P.: Front-end factor analysis for speaker verification. IEEE Trans. Audio Speech Lang. Process. **19**(4), 788–798 (2010)

9. Desplanques, B., Thienpondt, J., Demuynck, K.: ECAPA-TDNN: emphasized channel attention, propagation and aggregation in TDNN based speaker verification. In: Meng, H., Xu, B., Zheng, T.F. (eds.) Interspeech 2020, pp. 3830–3834. ISCA (2020)
10. Dutta, K., Mishra, J., Pati, D.: An effective combination scheme for improving speaker verification performance. In: TENCON 2017–2017 IEEE Region 10 Conference, pp. 1296–1299. IEEE (2017)
11. Dutta, K., Mishra, J., Pati, D.: Effective use of combined excitation source and vocal-tract information for speaker recognition tasks. Int. J. Speech Technol. **21**(4), 1057–1070 (2018)
12. Hermansky, H.: Perceptual linear predictive (PLP) analysis of speech. J. Acoust. Soc. Am. **87**(4), 1738–1752 (1990)
13. Kinnunen, T., Li, H.: An overview of text-independent speaker recognition: from features to supervectors. Speech Commun. **52**(1), 12–40 (2010)
14. Mishra, J., Singh, M., Pati, D.: LP residual features to counter replay attacks. In: 2018 International Conference on Signals and Systems (ICSigSys), pp. 261–266. IEEE (2018)
15. Mishra, J., Singh, M., Pati, D.: Processing linear prediction residual signal to counter replay attacks. In: 2018 International Conference on Signal Processing and Communications (SPCOM), pp. 95–99. IEEE (2018)
16. Murty, K.S.R., Yegnanarayana, B.: Combining evidence from residual phase and MFCC features for speaker recognition. IEEE Signal Process. Lett. **13**(1), 52–55 (2005)
17. Nassif, A.B., Shahin, I., Attili, I., Azzeh, M., Shaalan, K.: Speech recognition using deep neural networks: a systematic review. IEEE Access **7**, 19143–19165 (2019)
18. Prasanna, S.M., Gupta, C.S., Yegnanarayana, B.: Extraction of speaker-specific excitation information from linear prediction residual of speech. Speech Commun. **48**(10), 1243–1261 (2006)