Indonesian Journal of Electrical Engineering and Computer Science
Vol. 35, No. 2, August 2024, pp. 867~877
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v35.i2.pp867-877

Improving Kui digit recognition through machine learning and data augmentation techniques

Subrat Kumar Nayak[1], Ajit Kumar Nayak[2], Smitaprava Mishra[2], Prithviraj Mohanty[2], Nrusingha Tripathy[1], Sashikanta Prusty[1]
[1]Department of Computer Science and Engineering, Siksha 'O' Anusandhan (deemed to be University), Bhubaneswar, India
[2]Department of Computer Science and Information Technology, Siksha 'O' Anusandhan (deemed to be University), Bhubaneswar, India

Article Info

Article history:

Received Jan 24, 2024
Revised Mar 22, 2024
Accepted Apr 6, 2024

Keywords:

Data augmentation
Kui dataset
Low resource language
Mel-frequency cepstral coefficients
Speech recognition

ABSTRACT

Speech digit recognition research is growing decisively, and a bulk of digit recognition algorithms are used in European and a few Asian languages. Kui is a low-resourced tribal language locally used in several states of India. Despite its significance, there is not much research on Kui's speech. This research aims to present an in-depth analysis of novel Kui digit recognition using predefined machine learning (ML) techniques. For this purpose, we first gathered spoken numbers i.e. from 0 to 9 of eight different speakers containing a total of 200 words. Secondly, we choose the numbers: ଶୂନ (zero), ଏକ (one), ଦୁଇ (two), ତିନି (three), ସାରି (four), ପାସ (five), ସଅ (six), ସାତ (seven), ଆଟ (eight), ନଅ (nine). Meanwhile, we build nine different ML models to recognize Kui digits that take the Mel-frequency cepstral coefficients (MFCCs) method to extract the relevant features for model predictions. Finally, we compared the performance of ML models for both augmented and non-augmented Kui data. The result shows that the SVM+Augmentation method for Kui digit recognition combined obtained the highest accuracy of 83% than other methods. Moreover, the difficulties and potential prospects for Kui digit recognition are also highlighted in this work.

Corresponding Author:

Sashikanta Prusty
Department of Computer Science and Engineering, Siksha 'O' Anusandhan (deemed to University)
Bhubaneswar, India
Email: sashi.prusty79@gmail.com

1. INTRODUCTION

Computer science, probability theory, and optimization theory combine to form the discipline of machine learning (ML), which enables the completion of laborious tasks for which a logical or procedural approach would be impractical or impossible. The problem assembles all of the input data. It could be data gathered from devices, such as computers linked to the same network or a separate network, or it might be output from one program used as input by another program. The field of ML's most challenging problem right now is digit recognition. In this research, the topic is related to ML techniques [1]. Applying ML techniques to various audio datasets may resolve many practical issues. This study is based on a new dataset of Kui digits. Due to the absence of stemming, tokenizers, and significant publicly available corpora, Kui is still regarded as a low-resource language [2]. Training models for the Kui language is still difficult.

Kui is a tribal language that is mostly spoken in the tribal districts of Odisha [3]. Kui word datasets are less prevalent than other forms of online uploaded datasets. Kui word datasets are extremely uncommon and difficult to find. Compared to other languages, there is not a lot of study on the Kui language. Kui words are very rare as compared to other words, as they have no script. Kui words are written in Odia script. Any language cannot be researched or developed without data. No work has been done on spoken digit representation for Kui. Researchers need to create a robust and dependable speech recognition growth dataset, as no dataset is available to recognize Kui audio digits. This platform may typically add voice recognition in other low-resource languages [4]. However, many researchers developed several ML-based techniques for digit recognition. To boost performance, greater focus is necessary.

The main contributions of this paper are to:
- Gather a comprehensive Kui audio digit dataset in response to the need for Kui digit recognition. Digit recognition may be highly helpful in a wide variety of application areas. As with biometrics, the digit recognition system may be used as an authentication tool for security purposes.
- Implement the Mel-frequency cepstral coefficients (MFCC) method for extracting features during the preprocessing stage [5].
- Build ML models for voice digit recognition (i.e. Kui digits), based on past research outcomes on speech data.
- Design an algorithm to define our proposed model process workflow for Kui digit recognition using four necessary steps.
- Compare the performance of our proposed model using both augmented and without augmented Kui data.
- Design a comparison table describing the performance of our proposed method concerning some previous research done so far in this area.

The structure of the article is as follows. Comparing results and pertinent research is covered in Section 2. In section 3, the creation of the Kui dataset, feature extraction, and classification methodology are covered. In section 4, the suggested model and experimental configuration are described. Section 5 presents the findings. The conclusion and future directions of the work are presented in section 6.

2. RELATED WORK

Due to the rapid proliferation of internet information, digitization has emerged as one of the most crucial techniques for managing and organizing audio data. For the Kui language, very limited research has been done. Some Kui commands are trained and tested using a deep learning (DL) approach, which gives significant results [6]. Numerous research have been done to identify numbers from speech in various languages, but none have been done for Kui speech. The summary of Recognition of the digits using ML or DL techniques for different languages is shown in Table 1.

Table 1. Digit recognition using different learning techniques

Digit	Year of research	Digit	Year of research	Digit	Year of research
Arabic [7]	2004	Bangla [8]	2009	Malayalam [9]	2009
Kannada [10]	2012	Portuguese [11]	2012	Bodo [12]	2013
English [13]	2013	Gujarati [14]	2014	Urdu [15]	2015
Myanmar [16]	2016	Hindi [17]	2017	Isarn [18]	2017
Odia [19]	2019	Khasi [20]	2022	Algerian [21]	2022
Moroccan [21]	2022	Swahili [22]	2023	Kui	2024

3. MATERIALS AND METHODS

3.1. Data creation

This work created a Kui words dataset by collecting spoken words from different speakers of the Kandhamal District of Odisha. A platform was created for the Kui language's data preparation. The screenshot of the platform is shown in Figure 1. It takes longer to prepare the dataset. We have chosen different words. Table 2 displays the Kui digits' meanings.

The dataset consists of 1,600 utterances from 8 speakers. At a sampling rate of 16 kHz, each file is stored in wave format as linear 16-bit, single-channel phase change material (PCM) values. A Zoom audio recorder, a mobile device, and a laptop were used to record the audio in Kui. To reduce noise, it was recorded in a studio. The ratio of male to female voices in our sample is equal. The sample rate of each recording rate is examined following the conclusion of data collection.

3.2. Feature extraction techniques employed: MFCC

The feature extraction techniques are primarily helpful for Kui digit recognition. The majority of what it does is extract characteristics from audio sources. The retrieved features are sent as input into the classifier following feature extraction. The feature extraction phase of a speech conversion system is the first and most important. The discovery of a voice signal's specifics is a key objective of this approach. It is crucial to distinguish one speech from another. The extraction of characteristics needs to be constant throughout time. In speaking, it must happen regularly and naturally. In Figure 2, the MFCC block diagram is described. The following describes various methods for calculating the Mel-frequency cepstral coefficient [23].

Figure 1. Kui words data collection platform

Table 2. Kui digits and their meaning

Words (Kui)	Words (English)	Words (IPA)	Words (Roman)
ଶୂନ	Zero	sunɔ	suna
ଏକ	One	ekɔ	eka
ଦୁଇ	Two	d̪ui	dui
ତିନି	Three	t̪ini	tini
ସାରି	Four	t͡ʃa:ri	chāri
ପାସ	Five	pa:ɲt͡ʃɔ	pāñcha
ସଅ	Six	t͡ʃʰɔɔ	chhaa
ସାତ	Seven	sa:t̪ɔ	sāta
ଆଟ	Eight	a:ʈʰɔ	āṭha
ନଅ	Nine	nɔɔ	naa

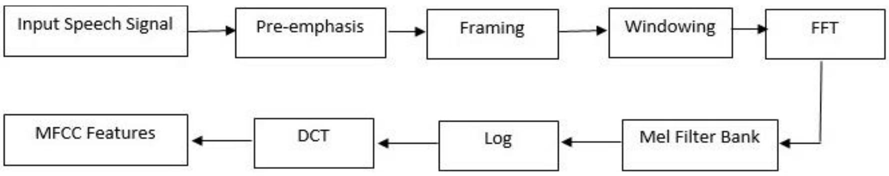

Figure 2. Block diagram of mel-frequency cepstral coefficient

3.3. Classification techniques

Several machine-learning approaches have been applied to voice digit recognition [24]. No work has yet been found for digit recognition in a low-resource language like Kui. For this research, we used several ML models. Applying the models, we compared the models' accuracy and performance metrics.

3.3.1. Adaptive boosting (AdaBoost)

Initialization: reset the weights to zero for every training example: $\omega_i^{(1)} = \frac{1}{N}$, where N is the number of training examples. For $t = 1$ to T (number of weak learners): compute learner weight: compute the weight of the weak learner:

$$\alpha_t = \frac{1}{2}\ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right) \quad (1)$$

update the weights of the training examples:

$$w_i^{(t+1)} = w_i^{(t)} \cdot exp\left(-\alpha_t \cdot y_i \cdot h_t(x_i)\right)$$

combine weak learners: the final strong learner prediction is a weighted sum of the weak learners' predictions:

$$H(x) = sign(\sum_{t=1}^{T} \alpha_t \cdot h_t(x)) \quad (2)$$

in the context of speech processing, the features x_i represent the characteristics extracted from the speech signals, and y_i represents the true label of whether the corresponding example is speech or non-speech. The weak learners($h_t(x)$) could be decision stumps that focus on specific speech features.

3.3.2. Decision tree

- Recursive node splitting: select best split: at each node, choose the feature F and, a threshold T to split the data into two subsets, D_{left} and, D_{right}, in a way that minimizes impurity or maximizes information gain:

$$Impurity(node) = ComputeImpurity(D_{node}) \quad (3)$$

- Prediction: given a new example with features i) X_{new} traverse the tree: start at the root and traverse down the tree by comparing the feature X_{new} with the node's splitting condition and ii) leaf node prediction: when a leaf node is reached, predict the majority class in that leaf node as the final prediction.

$$If\ X_{new}[F] \leq T\ then\ go\ left\ else\ go\ right \quad (4)$$

3.3.3. Linear discriminant analysis

- Training: compute class means: calculate the mean vector for each class. $\mu_k = \frac{1}{N_k}\sum_{i=1}^{N_k} X_i$, where N_k is the number of examples in class. k compute scatter matrices: within-class scatter matrix (S_W) as:

$$S_W = \sum_{k=1}^{C}\sum_{i=1}^{N_k}(X_i - \mu_k)\,(X_i - \mu_k)^T \quad (5)$$

- Prediction: for a new, unseen example, extract its features and project them onto the same LDA subspace as in (6). The trained classifier is then used to predict the class label for the new example based on the transformed features.

$$X_{new,lda} = X_{new} \cdot W \quad (6)$$

3.3.4. Support vector machine

- Training: for a linear SVM, the decision boundary (hyperplane) is represented as $\omega \cdot X + b = 0$ where ω is the weight vector and b is the bias. The optimization problem can be formulated as:

$$min_{w,b}\frac{1}{2}\|\omega\|^2 \text{subject to } Y_i(\omega \cdot X_i + b) \geq 1\ for\ i = 1,2, \dots., N \quad (7)$$

- Prediction: given a new speech example with features X_{new} , classify it based on the sign of $\omega \cdot\ X_{new} + b$.

$$Predicted\ Class =\ sign(\omega \cdot X_{new} + b) \quad (8)$$

3.3.5. Logistic regression

a. Training

- Sigmoid function: the logistic function is used to model the probability of the positive class: $P(Y = 1) = \frac{1}{1+e^{(\beta_0+\beta_1 X_1+\beta_2 X_2+\cdots+\beta_n X_n)}}$, where e the base of the natural logarithm is, β_0 is the intercept, and $\beta_1, \beta_2, \cdots, \beta_n$are the coefficients associated with the features as:

$$X_1, X_2, \dots, X_n. \quad (9)$$

- Log-Odds (logit) transformation: the logistic function can be transformed to express the log-odds (logit) as a linear combination of the features:

$$log\left(\frac{P(Y=1)}{1-P(Y=1)}\right) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_n \quad (10)$$

- Cost function (log-likelihood): the cost function to be minimized is the negative log-likelihood (cross-entropy loss) for binary classification.

$$J(\beta) = -\frac{1}{N}\sum_{i=1}^{N}[y_i log(p_i) + (1 - y_i) log(1 - p_i)]. \quad (11)$$

where N is the number of examples, y_i is the true label (0 or 1), and p_i is the predicted probability of the positive class.

b. Prediction: given a new example with features X_{new} , calculate the probability $P(Y = 1)$ using the trained coefficients and the logistic function.

3.3.6. Multilayer perceptron

- Training: Let L be the number of layers in the network, $N^{(l)}$be the number of neurons in layer l, and $W^{(l)}$ be the weight matrix connecting layer l to layer $l + 1$. The activation function $a^{(l)}$ is applied to the output of each neuron in layer l. The output of each neuron in the network is computed as follows:

$$a_i^{(l+1)} = g\left(\sum_{j=1}^{N^{(l)}} W_{ij}^{l} \cdot a_j^{(l)} + b_i^{(l)}\right), \quad (12)$$

where g is the activation function, W_{ij}^{l} is the weight connecting neuron j in layer l to neuron i in layer $+1$, and $b_i^{(l)}$ is the bias term for neuron i in layer $l + 1$.

- Prediction: given a new speech example with features X_{new} perform forward propagation through the trained network to obtain the predicted output.

3.3.7. Nearest centroid (NC)

- Training: For each class k, calculate the centroid (mean vector) μ_k of the feature vectors belonging to that class:

$$\mu_k = \frac{1}{N_k}\sum_{i=1}^{N_k} X_i^k \text{ , where } N_k \text{ is the number of examples in class } k. \quad (13)$$

- Prediction: Given a new speech example with features X_{new} , calculate the Euclidean distance to each class centroid, and assign the example to the class with the closest centroid.

$$Predicated\ Class = argmin_k \|X_{new} - \mu_k\|_2, \quad (14)$$

where $\|X_{new} - \mu_k\|_2$ represents the Euclidean distance between the features vector of the new example and the centroid of class k.

3.3.8. Random forest

- Training: Let D be the training dataset with N examples, each represented by a feature vector X_i and a class label Y_i in the case of classification. For $t = 1$ to T (number of trees): randomly select a bootstrap sample D_t from D with replacement. The size of D_t is N, but some examples may be repeated, and some may be left out. Randomly select m features from the total M features. Typically, $m \ll M$. Grow a decision tree T_t on the subset D_t using the selected features. Repeat this process until the tree is fully grown, potentially introducing additional randomness.
- Prediction: given a new speech example with features X_{new} , For each tree T_t in the forest, obtain the prediction $\hat{Y}_{new,t}$ based on the features of X_{new}. For classification tasks, the final predicted class is determined as follows.

$$\hat{Y}_{new} = argmax_c \sum_{t-1}^{T} \pi\left(\hat{Y}_{new,t} = c\right) \quad (15)$$

3.3.9. XGBoost

- Training: The training process in XGBoost involves optimizing an objective function that includes both a loss term and a regularization term:

$$Objective = \sum_{i=1}^{N} loss\left(Y_i, \hat{Y}_i\right) + \sum_{j=1}^{J} \Omega\left(f_j\right), \quad (16)$$

where N is the number of examples, $loss$ is the loss function measuring the difference between predicted $\left(\hat{Y}_i\right)$ and true labels (Y_i), J is the number of trees, and $\Omega\left(f_j\right)$ is the regularization term penalizing the complexity of each tree.

- Prediction: given a new speech example with features X_{new}, the final prediction is the sum of predictions from all trees:

$$\hat{Y}_{new} = \sum_{t=1}^{T} f_t\,(X_{new}), \text{ where } T \text{ is the total number of trees in the ensemble} \quad (17)$$

4. PROPOSED MODEL
4.1. Digit recognition system working model
As Kui is low-resourced it is very challenging to collect the data from the field. After the collection of data, we apply different machine learning models. The steps involved in this proposed Kui digit recognition system are depicted in Algorithm 1. The detailed workflow is presented in Figure 3.

Algorithm 1. Digit recognition system model
1. The voice data must be provided in the first phase. Here, we input 200 voice data for a single speaker. Pre-processing the input data provided in the first phase. The voice data is cleaned up and processed in this stage. This stage improves the quality of voice data for digit recognition.
2. Following the preceding phase, several prosodic or acoustic speech features are obtained from voice data. We use this technique to extract characteristics like pitch, energy, and intensity from speech data through MFCC, one of the feature extraction methods for voice data.
3. The ML models are used in the following phase to identify the digits.
4. The final phase is to measure the effectiveness of digit recognition using several metrics like precision, recall, and F1 score. It also calculated the accuracy of the ML models.

Figure 3. Kui digit recognition workflow

4.2. Data augmentation
Getting further information for less widely spoken languages might be difficult. Data augmentation is a technique used to increase the amount of data needed for training voice recognition systems. It is also a practical method for enhancing the present data's accessibility and enabling model training without the need for additional data. We have two options for displaying the audio data: raw audio and spectrogram [25]. This research uses noise injection and shifting time data augmentation techniques. After applying these techniques, the no of speakers increases to 24. Using the augmentation technique, we generate fresh, slightly modified samples from the original data to improve the training set. It might be viewed as a particular regularization strategy.

5. RESULTS AND DISCUSSION
However, so far there have been many researches carried out for digit recognition in different languages using ML techniques, but not in the case of Kui language. During the previous studies, researchers have faced many challenges like dataset preparation, preprocessing, and so on, some of which are discussed in related works. Furthermore, in earlier studies, difficulties arise when developing the dataset, especially in a tribal language as is not very easy, and also, the augmentation methods have neither applied nor compared to the original digit recognition dataset. To overcome such issues, this study briefly discusses dataset development in the case of a low-resourced tribal language and makes predictions with ML models. Moreover, we compared the performance of ML models using augmentation and without augmentation methods to get the best predictive one.

The performance measures for the predictive model include precision, recall, F1-Score, and accuracy. These performance parameters are determined, as in (1)-(4), based on the obtained confusion matrix (CM). Each confusion matrix has a y-axis that displays the actual labels and an x-axis that displays the predicted labels. The CM for several ML models, both with and without data augmentation, is shown in Figures 4 to 21.

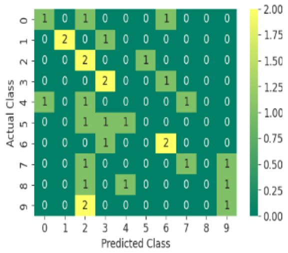

Figure 4. CM obtained from AdaBoost without augmentation

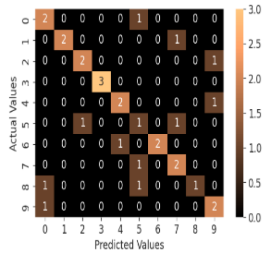

Figure 5. CM obtained from AdaBoost with augmentation

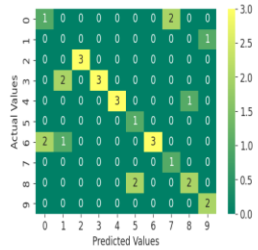

Figure 6. CM obtained from LR without augmentation

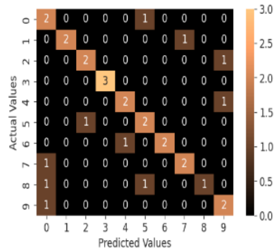

Figure 7. CM obtained from LR with augmentation

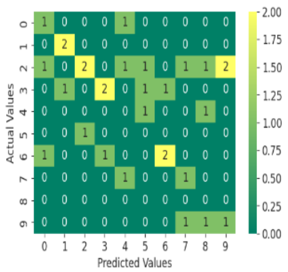

Figure 8. CM obtained from DT Without augmentation

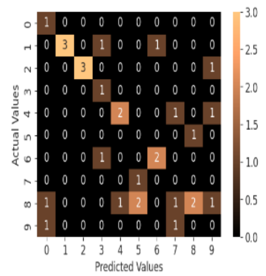

Figure 9. CM obtained from DT with augmentation

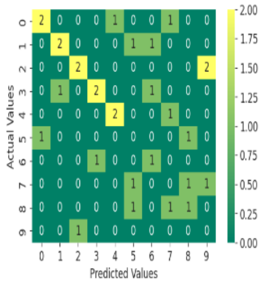

Figure 10. CM obtained from MLP without augmentation

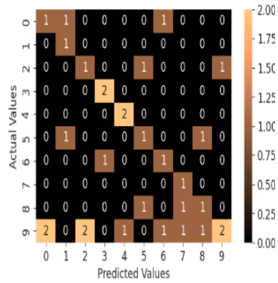

Figure 11. CM obtained from MLP with augmentation

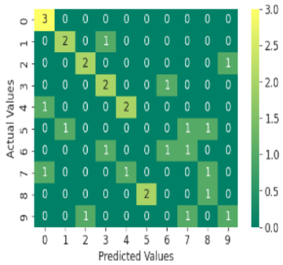

Figure 12. CM obtained from the LDA without augmentation

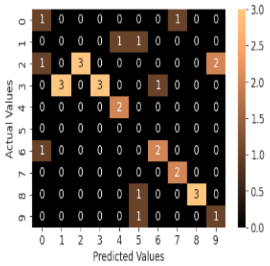

Figure 13. CM obtained from the LDA with augmentation

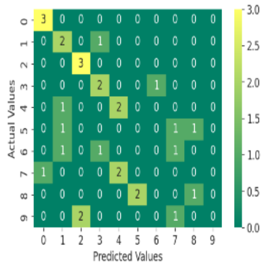

Figure 14. CM obtained from NC without augmentation

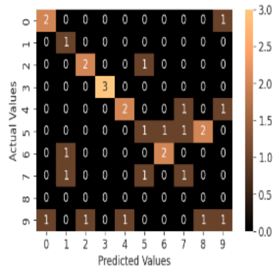

Figure 15. CM obtained from NC with augmentation

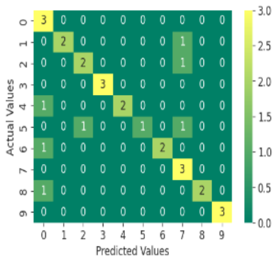

Figure 16. CM obtained from the SVM without augmentation

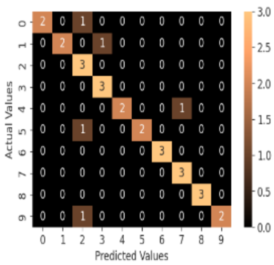

Figure 17. CM obtained from the SVM with augmentation

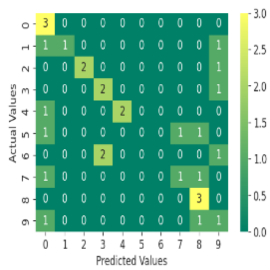

Figure 18. CM obtained from RF without augmentation

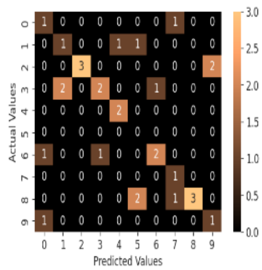

Figure 19. CM obtained RF with augmentation

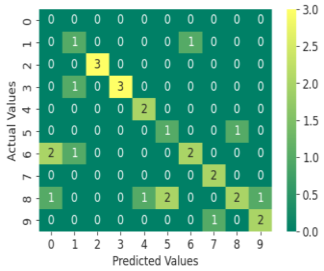

Figure 20. CM obtained from XGBoost without augmentation

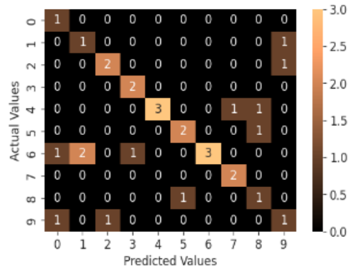

Figure 21. CM obtained from XGBoost with augmentation

As previously mentioned, we assess ML models using the Kui dataset. Thirty-two batches are allowed for every training session. First, we train the model without data augmentation using different parameters when learning rate=0.01. The most popular and straightforward performance evaluation parameter for digit recognition is accuracy. The recall is the proportion of all instances properly classified as belonging to the positive class to all actual members of the positive class. F1-Score is dependent on the model's recall and accuracy. Tables 3 and 4 provides information without and with augmentation on testing accuracy respectively. Then, the accuracy of different ML models is shown in Figure 22.

The performance matrix of SVM gives a significant result compared to others using our Kui dataset, shown in Figures 16 and 17. The accuracy of different ML models in different languages is shown in Table 5. It shows that Kui digit recognition gives the highest accuracy compared to the other languages using the SVM classifier.

Table 3. Comparison of nine different ML model's performance before augmentation

ML models	Performance metric			
	Precision	Recall	F1-Score	Accuracy
AdaBoost	45	48	46	47
DT	52	37	40	37
LDA	44	47	45	47
LR	79	63	67	63
MLP	48	40	43	40
NC	33	43	36	43
RF	51	50	45	50
XGBoost	70	60	62	60
SVM	87	77	77	77

Table 4. Comparison of nine different ML model's performance after augmentation

ML models	Performance metric			
	Precision	Recall	F1-Score	Accuracy
AdaBoost	77	57	61	57
DT	66	47	51	47
LDA	56	51	48	50
LR	74	67	67	67
MLP	49	43	40	43
NC	54	50	51	50
RF	70	53	57	53
XGBoost	71	60	61	60
SVM	90	83	84	83

Table 5. Accuracy comparison of different models for different languages

Work	Digit	Methods	Year of experiment	Performance evaluation (In %)			
				Precision	Recall	F1-Score	Accuracy
Hegde *et al.* [10]	Kannada	SVM	2012	-	-	-	79
Thakuria *et al.* [12]	Bodo	HMM	2013	-	-	-	70
Ali *et al.* [15]	Urdu	SVM	2015	-	-	-	73
Tun *et al.* [16]	Myanmar	HMM	2016	-	-	-	53
Sangjamraschaikun *et al.* [18]	Isarn	HMM	2017	-	-	-	79
Syamanthika *et al.* [26]	Tamil	SVM	2020	51	46	41	47
Proposed Method	Kui	SVM with data augmentation	2024	90	83	84	83

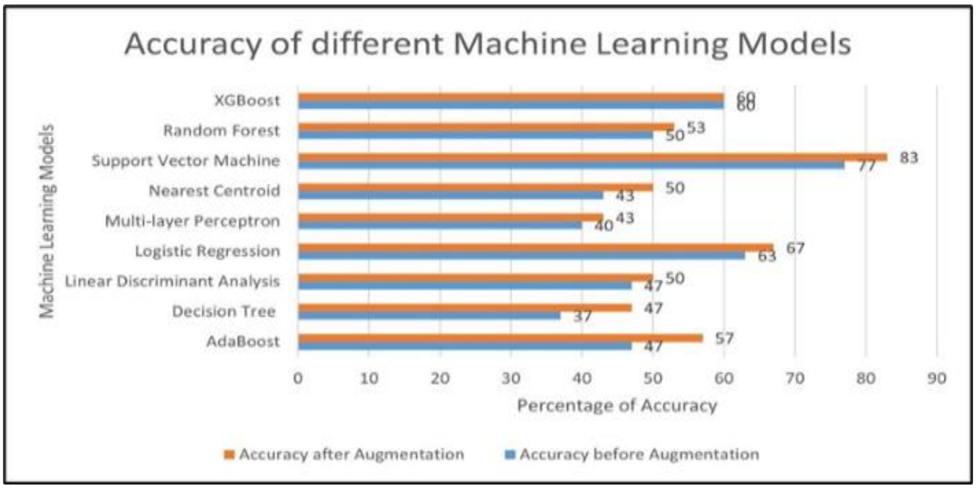

Figure 22. Accuracy of different ML models

Besides the advantage of using the SVM + augmentation method for Kui digit recognition, we have faced some challenges with a very small number of speakers participating in the case of our developed Kui dataset. Secondly, the intergenerational transmission of these languages reduces as younger generations migrate toward more prevalent languages, which further reduces the population of speakers. Thirdly, the Kui language has primarily been oral traditions, with no written form. It may be difficult to integrate the language into formal education systems or to maintain it for future generations.

4. CONCLUSION AND FUTURE WORK

Digit recognition is widely used across industries for various applications, including security. In this research, the real-time Kui digit dataset has collected. The highest performance is achieved using the SVM with data augmentation technique, i.e., 83%, than other models. Adding more Kui terms to the dataset and training the model can fill certain gaps in our work. Due to the enormous number of people who speak Kui, it is important to diversify the Kui speech corpus when constructing or gathering data for Kui digit recognition. As a result, the Kui corpus will be diverse in terms of dialectal and accent information. We are concentrating on dataset diversity and quantity, as digit recognition requires a variety of speakers to hear more and more varied voices. Multiple speakers using the same voice will not aid in determining digit recognition. The accuracy of the findings will significantly increase with further development of the algorithms, the training dataset, and the classifier itself. In the future, we intend to extend our dataset size, extract the features using different feature extraction methods, and compare the result by applying different classifiers.

REFERENCES

[1] M. Mohammed, M. B. Khan, and E. B. M. Bashie, *Machine learning: Algorithms and applications*. CRC Press, 2016. doi: 10.1201/9781315371658.

[2] T. Burrow, "Some notes on the Kui dialects spoken by the Kuttiga Kondhs of North East Koraput," *Indo-American Journal*, pp. 118–35, 1961.

[3] Kutia Kondh. (Tribal language series), Bhubaneswar: Academy of Tribal Dialect and Culture, 2001.

[4] "Language Corpus Collection," janabhasha.in. https://janabhasha.in/bhashasangraha/index.php (accessed Oct. 17, 2022).

[5] S. K. Nayak *et al.*, "Speech data collection system for KUI, a Low resourced tribal language," *Journal of Autonomous Intelligence*, vol. 7, no. 1, Oct. 2023, doi: 10.32629/jai.v7i1.1121.

[6] S. K. Nayak, A. K. Nayak, S. Mishra, and P. Mohanty, "Deep learning approaches for speech command recognition in a Low Resource KUI Language," *International Journal of Intelligent Systems and Applications in Engineering*, vol. 11, no. 2, pp. 377–386, 2023.

[7] R. Djemili, M. Bedda, and H. Bourouba, "Recognition of spoken Arabic digits using neural predictive hidden markov models," *The International Arab Journal of Information Technology*, vol. 1, no. 2, 2004.

[8] G. Muhammad, Y. A. Alotaibi, and M. N. Huda, "Automatic speech recognition for Bangla digits," in *2009 12th International Conference on Computers and Information Technology*, IEEE, Dec. 2009, pp. 379–383. doi: 10.1109/ICCIT.2009.5407267.

[9] C. Kurian and K. Balakrishnan, "Speech recognition of Malayalam numbers," in *2009 World Congress on Nature and Biologically Inspired Computing, NABIC 2009 - Proceedings*, IEEE, 2009, pp. 1475–1479. doi: 10.1109/NABIC.2009.5393692.

[10] S. Hegde, K. K. Achary, and S. Shetty, "Isolated word recognition for Kannada language using support vector machine," in *Communications in Computer and Information Science*, vol. 292 CCIS, 2012, pp. 262–269. doi: 10.1007/978-3-642-31686-9_31.

[11] D. F. Silva, V. M. A. de Souza, G. E. A. P. A. Batista, and R. Giusti, "Spoken digit recognition in portuguese using line spectral frequencies," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 7637 LNAI, 2012, pp. 241–250. doi: 10.1007/978-3-642-34654-5_25.

[12] L. Thakuria, A. Das, P. Acharjee, and P. H. Talukdar, "Automatic speech recognition of BODO alpha digits using hidden morkov models," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 3, no. 11, 2013.

[13] V. Trivedi, "A survey on english digit speech recognition using HMM," *International Journal of Science and Research*, vol. 2, pp. 2319–7064, 2013, [Online]. Available: www.ijsr.net

[14] P. Pandit, S. Bhatt, and P. Makwana, "Automatic speech recognition of gujarati digits using artificial neural network," *Proceedings of 19th Annual Cum 4th International Conference of GAMS On Advances in Mathematical Modelling to Real World Problems*, pp. 141–146, 2014.

[15] H. Ali, A. Jianwei, and K. Iqbal, "Automatic speech recognition of urdu digits with optimal classification approach," *International Journal of Computer Applications*, vol. 118, no. 9, pp. 1–5, 2015, doi: 10.5120/20770-3275.

[16] Z. Z. Tun and G. Srijuntongsiri, "A speech recognition system for myanmar digits," *International Journal of Information and Electronics Engineering*, vol. 6, no. 3, pp. 210–213, 2016, doi: 10.18178/IJIEE.2016.6.3.626.

[17] P. Prajapati and M. Patel, "A survey on isolated word and digit recognition using different techniques," *International Journal of Computer Applications*, vol. 161, no. 3, pp. 6–15, Mar. 2017, doi: 10.5120/ijca2017913130.

[18] S. Sangjamraschaikun and P. Seresangtakul, "Isarn digit speech recognition using HMM," in *Proceeding of 2017 2nd International Conference on Information Technology, INCIT 2017*, IEEE, Nov. 2017, pp. 1–5. doi: 10.1109/INCIT.2017.8257882.

[19] P. Mohanty and A. K. Nayak, "Isolated spoken odia digit recognition using support vector machine," in *Proceedings - 2019 International Conference on Applied Machine Learning, ICAML 2019*, IEEE, May 2019, pp. 147–152. doi: 10.1109/ICAML48257.2019.00036.

[20] F. Rynjah, B. Syiem, and L. J. Singh, "Speech recognition system of spoken isolated digit in standard khasi dialect," *Lecture Notes in Networks and Systems*, vol. 404, pp. 541–549, 2023, doi: 10.1007/978-981-19-0105-8_53.

[21] K. Lounnas, M. Abbas, M. Lichouri, M. Hamidi, H. Satori, and H. Teffahi, "Enhancement of spoken digits recognition for under-resourced languages: case of Algerian and Moroccan dialects," *International Journal of Speech Technology*, vol. 25, no. 2, pp. 443–455, 2022, doi: 10.1007/s10772-022-09971-y.

[22] A. R. Kivaisi, Q. Zhao, and J. T. Mbelwa, "Swahili speech dataset development and improved pre-training method for spoken digit recognition," *ACM Transactions on Asian and Low-Resource Language Information Processing*, vol. 22, no. 7, pp. 1–24, Jul. 2023, doi: 10.1145/3597494.

[23] S. Prusty, S. Patnaik, and S. K. Dash, "Comparative analysis and prediction of coronary heart disease," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 27, no. 2, p. 944, Aug. 2022, doi: 10.11591/ijeecs.v27.i2.pp944-953.

[24] N. Tripathy, S. Hota, and D. Mishra, "Performance analysis of bitcoin forecasting using deep learning techniques," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 31, no. 3, p. 1515, Sep. 2023, doi: 10.11591/ijeecs.v31.i3.pp1515-1522.

[25] H. Bakır, A. N. Çayır, and T. S. Navruz, "A comprehensive experimental study for analyzing the effects of data augmentation techniques on voice classification," *Multimedia Tools and Applications*, vol. 83, no. 6, pp. 17601–17628, Aug. 2023, doi: 10.1007/s11042-023-16200-4.

[26] P. Syamanthika, T. Yogitha, M. K. Sai Hitha, T. M. Swetha, S. Poorna, and K. Anuraj, "Digit Identification from Speech using Short-Time Domain Features," in *2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA)*, IEEE, Jul. 2020, pp. 84–87. doi: 10.1109/ICIRCA48905.2020.9182788.

BIOGRAPHIES OF AUTHORS

Subrat Kumar Nayak received the degree in MCA from Biju Patnaik University of Technology, Odisha, India in 2010, M. Tech in Computer Science from Utkal University, Bhubaneswar, Odisha in 2012. He is currently pursuing his Ph.D. in CSE at SOA University, Bhubaneswar, India. He has published 7 papers in various International Journals and International conferences. He qualified for UGC Net in the year 2012. He has more than 6 years of academic experience and 3 years of govt experience. He can be contacted at email: subratsilicon28@gmail.com.

Prof. Dr. Ajit Kumar Nayak is the professor and Head of the Department of Computer Science and Information Technology, Siksha 'O' Anusandhan Deemed to be University, Bhubaneswar, Odisha. He received degree Electrical in Engineering from the Institution of Engineers, India in the year 1994, M. Tech and and Ph.D. degree in Computer Science from Utkal University in 2001 and 2010 respectively. He has published about 70 research papers in various journals and conferences. Also co-authored a book 'Computer Network Simulation using NS2'. Ten PhD. scholars have been awarded Ph.D. under his supervision. He can be contacted at email: ajitnayak@soa.ac.in.

Prof. Dr. Smitaprava Mishra is currently working as Professor in Department of Computer Science and Information Technology at Siksha 'O' Anusandhan Deemed to be University, Bhubaneswar, Odisha. She has 18years' of teaching and research experience in the current organization. She has published 30 numbers of research papers in various reputed journals and conferences. M.Tech graduates a total of 15 people were produced under his supervision in the research fields of data mining, machine learning, and natural language processing. The Ph.D. as many as 4 people conducted research under his supervision. She has contributed several academic activities at organization level. She can be contacted at email: smitamishra@soa.ac.in.

Dr. Prithviraj Mohanty is currently working as Associate Professor in Department of Computer Science and Information Technology at Siksha 'O' Anusandhan Deemed to be University, Bhubaneswar, Odisha. He received his Ph.D degree in CSE at Siksha 'O' Anusandhan University, Bhubaneswar, Odisha. He received his B.Tech. degree in CSE from Biju Patnaik University of Technology (BPUT), Odisha and M. Tech from KIIT University, Bhubaneswar. His research interest includes speech recognition, image processing, machine learning, artificial intelligence, grid computing and distributed computing. He can be contacted at email: prithvirajmohanty@soa.ac.in.

Nrusingha Tripathy received the MCA degree in Computer Science from Ravenshaw University, Cuttack, Odisha, India in 2018, and the M. Tech degree in CSE from Utkal University, Bhubaneswar, Odisha, India in 2020. He is currently pursuing his Ph.D. in CSE at Siksha O Anusandhan Deemed to be University, Bhubaneswar, India, and has published 5 Conferences. Moreover, 10 journal papers have been published. Although, he has 5+ years of teaching experience. He can be contacted at email: nrusinghatripathy654@gmail.com.

Sashikanta Prusty received an MCA degree in Computer Science & Engineering from PIET, Rourkela, Odisha, India in 2010. M. Tech degree in CSE from Raajdhani Engineering College, Bhubaneswar, Odisha, India in 2020. He has completed his Ph.D. in CSE at SOA University, Bhubaneswar, India, and has published 4 SCIE and 13 Scopus-indexed Journals. Also, published 12 IEEE conferences and 2 Springer book chapters. He has 4+ years of teaching and 3 years of industry experience. He can be contacted at email: sashi.prusty79@gmail.com.