International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Empowering Accented Speech Analysis in Malayalam Through Cutting-Edge Fusion of Self Supervised Learning and Autoencoders

Rizwana Kallooravi Thandil[1], Mohamed Basheer K. P.[2] and Muneer V. K.[3]

Submitted: 17/10/2023 **Revised**: 06/12/2023 **Accepted**: 15/12/2023

Abstract: This research explores the application of autoencoders in handling accented speech data for the Malayalam language. The primary objective is to leverage the power of autoencoders to learn a compressed representation of the input data and utilize it to train various machine learning models for improved accuracy rates and reduced word error rates (WER). The study involves a two-step process. Firstly, an autoencoder neural network architecture is employed to encode the accented speech data into a lower-dimensional latent space representation. The encoder network effectively captures the essential features and patterns present in the data. The compressed representation obtained from the encoder is then fed into the decoder, which reconstructs the original input data. In the second step, the encoded model is utilized to train several machine learning models, including logistic regression, decision tree classifier, support vector machine (SVM), random forest classifier(RFC), K-nearest neighbors (KNN), stochastic gradient descent (SGD), and multilayer perceptron (MLP). The encoded features act as inputs to these models, enabling them to learn from the compact representation of the accented speech data. Experimental results indicate that the trained machine learning models, using the encoded features, achieve higher accuracy rates compared to traditional approaches. This improvement in accuracy demonstrates the effectiveness of autoencoders in capturing and representing the significant characteristics of the accented speech data. Moreover, the utilization of the encoded model also leads to lower word error rates, indicating enhanced performance in accurately transcribing and recognizing accented speech in the Malayalam language. This finding showcases the potential of autoencoders in improving the overall accuracy and efficiency of speech-processing tasks for accented languages.

Keywords: *Autoencoders, self-supervised learning, human-computer interface, Accented speech recognition, Malayalam speech recognition*

1. Introduction

Accented speech analysis in the context of the Malayalam language poses significant challenges due to the wide range of accents present within the language. Accurate analysis and interpretation of accented speech are crucial for applications such as speech recognition, language understanding, and accent identification. In recent years, the combination of autoencoders, self-supervised learning, and machine learning models have emerged as a promising direction for improving accent speech analysis.

This research aims to contribute to the field of accent speech analysis in Malayalam by leveraging the power of autoencoders and machine learning models. Specifically, the focus is on exploring the potential of compressed autoencoder models for feature extraction and using the extracted features to train various machine learning algorithms, including logistic regression, decision tree classifier, SVM, random forest, KNN, SGD, and MLP.

The research investigates the effectiveness of the compressed autoencoder model in capturing essential patterns and features within the accented speech data. By compressing the data into a lower-dimensional latent space, the autoencoder extracts informative representa tions that effectively capture the variations in accents while minimizing the loss of relevant information.

Furthermore, this research employs self-supervised learning techniques to guide the training of the autoencoder model. Self-supervised learning leverages the intrinsic structure of the data to learn meaningful representations without requiring extensive labeled datasets. This approach allows for unsupervised representation learning, which is particularly advantageous in scenarios where labeled data is limited or unavailable.

By training the various machine learning models with the encoded representations obtained from the compressed autoencoder model, the research aims to enhance the accuracy and robustness of accent speech analysis. The evaluation and comparison of these models provide insights into their performance, strengths, and limitations in the specific context of Malayalam-accented speech. The

[1]*,[2,3]*Assistant Professor, Sullamussalam Science College, Areekode, Kerala, India.* [1]*Email: Ktrizwana@gmail.com*
[1]*[0000-0001-9305-681X],*
[2]*[0000-0003-3847-4124]*
[3]*[0000-0002-5517-2462]*

outcomes of this research have practical implications for applications such as speech recognition, accent identification, and language processing in the Malayalam language. By improving the accuracy rates and reducing Word Error Rates (WER) in accent speech analysis, the developed approach has the potential to advance the state-of-the-art systems in these domains, benefiting users and applications that rely on effective analysis and understanding of accented speech. This research bridges the gap between autoencoder models, self-supervised learning, and machine learning algorithms in the domain of accent speech analysis. The exploration of compressed autoencoder models and their integration with machine learning techniques contributes to the development of more accurate and robust systems for analyzing and interpreting accented speech in the context of Malayalam.

The key contributions of this work are :

1. Autoencoder Model Construction: The research involves the construction of an autoencoder model for accent speech analysis in Malayalam. This model serves as a powerful tool for unsupervised learning by capturing the underlying patterns and features within the data.
2. Analysis without Data Compression: One contribution of this work is exploring the effectiveness of the autoencoder model without compressing the data. This analysis allows for a comprehensive understanding of the accent speech data without any dimensionality reduction, providing insights into the raw representation of the features.
3. Analysis with Compressed Data: Another contribution is evaluating the performance of the autoencoder model when data compression is applied. This involves encoding the high-dimensional accent speech data into a lower-dimensional latent space representation, effectively reducing the dimensionality of the features while preserving relevant information.
4. Comparative Analysis: A comparative analysis between the results obtained from the autoencoder model with and without data compression. This comparison allows for an assessment of the benefits and trade-offs associated with compressing the accent speech data, such as the impact on accuracy, computational efficiency, and interpretability of the learned representations.
5. Improved Accuracy: The training of machine learning algorithms using the encoded representations obtained from the autoencoder model leads to higher accuracy rates in accent speech analysis. This improvement suggests that the compressed representations capture the essential information necessary for accurate classification or prediction tasks.
6. Lower WER: One of the key contributions of this work is the observation of higher accuracy rates and lower Word Error Rates (WER) in accent speech analysis compared to traditional approaches.
7. Practical Significance: The outcomes of this work have practical significance in the field of accent speech analysis. The improved accuracy rates and lower WER achieved through the utilization of the compressed autoencoder model and various machine learning algorithms contribute to the development of more reliable and efficient systems for applications such as speech recognition, language processing, and accent identification.

2. Related Work

Accented speech recognition poses unique challenges in accurately understanding and interpreting spoken language due to variations in pronunciation, intonation, and phonetic characteristics across different accents. Traditional methods for speech recognition often struggle to handle such variations, leading to degraded performance and reduced accuracy. Autoencoders offer a powerful approach to learning robust and discriminative representations directly from raw speech data, without the need for explicit feature engineering. It compresses the input speech signals into a lower-dimensional latent space and then reconstructs them. By training the autoencoder to minimize the reconstruction error, it learns to extract salient features that capture the essential information for accurate speech recognition.

Accented speech recognition using autoencoders involves training the autoencoder model on a diverse dataset that includes speakers with various accents. This enables the model to learn accent-invariant representations by capturing the underlying shared characteristics of speech, while also accounting for the specific accent variations. Some of the relevant works in the literature are discussed below.

Sahu et. al [1] in their work addresses the limitations of traditional feature extraction methods by leveraging the power of autoencoders to learn compact and discriminative representations directly from raw speech signals. They introduce an adversarial training framework where a generator network, implemented as an autoencoder, is pitted against a discriminator network.

Lee et. al[2] introduce a novel approach that leverages chain-based discriminative autoencoders for speech recognition. It highlights the benefits of incorporating contextual information and discriminative criteria in the autoencoder framework, leading to enhanced performance in speech recognition tasks.

Deng et. al [3] propose an approach to speech emotion recognition using semi-supervised autoencoders. The authors address the challenge of limited labeled data in emotion recognition tasks by leveraging unlabeled data to enhance the performance of the emotion recognition model. They propose a semi-supervised learning framework that combines the power of autoencoders with limited labeled data and a large amount of unlabeled data.

Karitha et. al[4] in their work present a semi-supervised learning approach that combines text-to-speech synthesis and autoencoders for ASR. It showcases the benefits of using synthesized data to augment the labeled data, enhancing the performance of the speech recognition model.

Huang et. al[5] in their work introduces masked autoencoders with attention mechanisms as a powerful tool for speech recognition tasks. By allowing the model to selectively attend to relevant acoustic segments, the proposed approach improves the model's ability to capture fine-grained details and enhances speech recognition performance.

Atmaja et. al[6] explores the potential of self-supervised learning techniques to learn informative representations from unlabeled data, leading to improved performance in emotion recognition tasks.

Peng et. al[7] present an autoencoder-based feature-level fusion technique by combining multiple acoustic features through autoencoder-based representations, the proposed approach effectively captures emotional information and improves the performance of SER systems.

Bastanfard et. al[8] present a stacked autoencoder-based approach for speech emotion recognition in the Persian language. By comparing local and global features, the study highlights the importance of considering different feature types in capturing emotional information from speech signals. The proposed method shows promising results in recognizing emotions from Persian speech data.

Ying et. al[9] propose an unsupervised feature learning approach using autoencoders for speech emotion recognition. By learning discriminative representations directly from raw speech signals, the proposed method eliminates the need for handcrafted features and achieves better performance in emotion classification tasks.

By effectively incorporating autoencoders, researchers, and practitioners are advancing the field of accented speech recognition, enabling more accurate and robust systems for speech analysis, transcription, and natural language understanding across diverse accents and languages.

3. Methodology

The entire study is conducted in nine steps and the steps involved are shown in Figure 1

Fig 1 The steps involved in the entire study

1. Data Collection

The speech corpus was carefully developed under natural recording conditions and comprised approximately 1.17 hours of accented speech. To ensure diversity and representation, data collection involved forty speakers, including twenty males and twenty females, from five distinct districts in Kerala, where Malayalam is spoken with different accents.

The selection of speakers aimed to encompass a wide range of ages, with participants ranging from five to eighty years old. The corpus construction process focused on capturing individual utterances of multi-syllabled words, with each sample lasting between two to five seconds. The targeted districts for data collection were Kasaragod, Kannur, Kozhikode, Wayanad, and Malappuram in Kerala. These districts were chosen due to the distinct accents influenced by the languages spoken in neighboring states that share borders with them. By developing this custom speech corpus, the authors aimed to address the scarcity of appropriate data for accented speech analysis.

2. Data Preprocessing

Cleaned and preprocessed the collected speech data to remove any artifacts, background noise, or irrelevant segments. We have applied the techniques such as noise reduction, filtering, and normalization to enhance the quality of the data.

3. Feature Extraction

In our research on accented speech recognition, we extract relevant acoustic features from preprocessed speech data to capture important characteristics such as pitch, tempo, spectral frequencies, and rhythm. Three key feature extraction techniques are employed: Mel-frequency cepstral coefficients (MFCCs), Short Term Fourier Transform (STFT), and Tempogram analysis. MFCC can be computed by the formula:

$$MFCCs(t) = DCT(\log(E(t) * H(t))) \quad (1)$$

where E(t) represents the magnitude spectrum of the preprocessed speech frame at time t, H(t) is the mel-filterbank matrix, and DCT denotes the Discrete Cosine Transform.

For our study, we consider the first 13 MFCC coefficients along with their first and second derivatives. These derivatives provide additional information about the spectral dynamics of the speech samples, capturing changes in the spectral content over time. The formulas for calculating the first and second derivatives of the MFCC coefficients are as follows:

$$MFCC'(t) = (MFCC(t+1) - MFCC(t-1)) / 2 \quad (2)$$

$$MFCC''(t) = MFCC(t+1) - 2 * MFCC(t) + MFCC(t-1) \quad (3)$$

To refine the MFCC representation, we calculate the mean of all 39 values (13 coefficients + 13 first derivatives + 13 second derivatives) and add it to the list of speech vectors. This step enhances the robustness and stability of the MFCC representation, resulting in a total of 40 MFCC vectors for each speech signal. Apart from MFCCs, we employ the STFT to analyze the temporal and spectral characteristics of speech signals. The STFT represents the speech signal in the time-frequency domain and is computed using the following formula:

$$STFT(t, f) = |s(t, f)|^2 \quad (4)$$

where s(t, f) denotes the spectrogram magnitude at time t and frequency f.

Furthermore, we utilize Tempogram analysis to capture the rhythmic patterns and accent variations in the speech signals. The formula for Tempogram computation is as follows:

$$Tempogram(t, f) = \sum[s(t, f) * s(t + \tau, f)] \quad (5)$$

where s(t, f) denotes the onset strength envelope of the speech signal at time t and frequency f, and τ represents the time lag. The Tempogram is calculated by applying the autocorrelation function to the onset strength envelope of the speech signal. It provides a representation that highlights the temporal changes in the speech rhythm, making it useful for characterizing accents and rhythmical nuances. By combining the MFCCs, STFT, and Tempogram features, we extract a total of 436 features for each speech sample. These features provide a comprehensive representation of the speech signals, encompassing crucial characteristics related to pitch, tempo, spectral frequencies, and rhythm. They serve as the foundation for subsequent analysis and classification tasks in the domain of accented speech recognition.

4. Autoencoder Training and Self-Supervised Learning Framework

We have designed and trained an autoencoder architecture using preprocessed speech data. The autoencoder comprises an encoder and a decoder. The encoder maps the high dimensional input features to a lower-dimensional latent space, capturing the essential information. The decoder reconstructs the input data from the latent space representation.

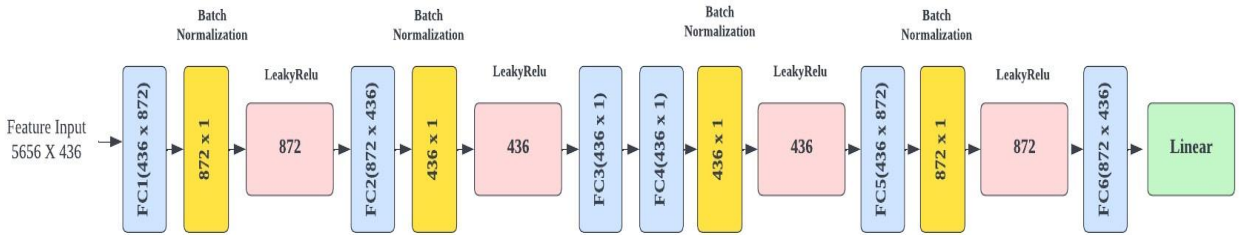

Fig 2 Autoencoder Model Architecture Without Compression

The architecture shown in Figure 2 encompasses an encoder network that takes the input data and passes it through a series of fully connected layers. The autoencoder consists of an input layer with X as the input, representing a set of 436 feature vectors. The first encoder layer performs a linear transformation followed by the LeakyReLU activation function, which can be calculated as

$$e = W\_1 * X + b\_1 \quad (6)$$

Batch normalization is then applied to e to normalize the outputs. The LeakyReLU activation function is applied element-wise to the normalized outputs.

The second encoder layer performs another linear transformation followed by the LeakyReLU activation function. It can be represented as

$$e = W\_2 * e + b\_2 \quad (7)$$

Batch normalization is applied to e to normalize the outputs, and the LeakyReLU activation function is applied element-wise.

The bottleneck layer has the same number of neurons as the number of input features. It performs a linear transformation, which can be represented as

$$bottleneck = W\_b * e + b\_b \quad (8)$$

The first decoder layer performs a linear transformation, d

$$= W\_d1 * bottleneck + b\_d1 \quad (9)$$

After applying batch normalization to variable 'd' to normalize the outputs, the LeakyReLU activation function is applied element-wise. Subsequently, the second decoder layer performs another linear transformation, given by the equation

$$d = W\_d2 * d + b\_d2 \quad (10)$$

Here, W_d2 represents the weight matrix, b_d2 represents the bias vector, and d denotes the output of the second decoder layer.

The output layer performs a linear transformation with a linear activation function, represented as

$$output = W\_out * d + b\_out \quad (11)$$

The weight matrices (W_1, W_2, W_b, W_d1, W_d2, W_out) and bias vectors (b_1, b_2, b_b, b_d1, b_d2, b_out) are learned during the training process and are used to perform the respective linear transformations in the network.

In the second phase, we constructed the autoencoder model with compressed data. The first encoder layer performs a linear transformation followed by batch normalization and LeakyReLU activation element-wise. which can be represented as:

$$e = W_1 * F + b_1 \quad (12)$$

where W_1 is the weight matrix, F is the input feature set, b_1 is the bias vector, and e is the output.

The second encoder layer performs another linear transformation followed by batch normalization and LeakyReLU activation element-wise which can be represented as $e = W_2 * e + b_2$ (13)

where W_2 is the weight matrix and b_2 is the bias vector, and e is the output.

The bottleneck layer performs a linear transformation on the outputs of the second encoder layer which can be represented as:

$$bottleneck = W\_b * e + b\_b \quad (14)$$

where W_b is the weight matrix and b_b is the bias vector, and the bottleneck is the output.

The first decoder layer performs a linear transformation followed by batch normalization and LeakyReLU activation element-wise and can be represented as:

$$d = W\_d_1 * bottleneck + b\_d \quad (15)$$

where W_d_1 is the weight matrix and b_d_1 is the bias vector, and d is the output.

The second decoder layer performs another linear transformation followed by batch normalization and LeakyReLU activation element-wise. Mathematically, it can be represented as:

$$d = W\_d_2 * d + b\_d_2 \quad (16)$$

where W_d_2 is the weight matrix and b_d_2 is the bias vector, and d is the output.

The output layer performs a linear transformation with a linear activation function. Mathematically, it can be represented as:

$$output = W\_out * d + b\_out \quad (17)$$

where W_out is the weight matrix and b_out is the bias vector, and output is the final output.

Here in this architecture, the encoder is composed of two dense layers with leaky ReLU activation and batch normalization. The first dense layer has twice the number of neurons as the input features, and the second dense layer has the same number of neurons as the input features. These layers gradually reduce the dimension- nality of the data and capture meaningful representations. The bottleneck layer, which is the output of the second dense layer, has half the number of neurons as the input features. It serves as the compressed representation of the input data.

The decoder is constructed as the reverse of the encoder architecture. It also consists of two dense layers with leaky ReLU activation and batch normalization. The output layer has the same number of neurons as the input features and uses a linear activation function. During training, the autoencoder aims to reconstruct the input data by minimizing the difference between the input and output. The training is performed for 500 epochs with a batch size of 16. Additionally, an encoder model is defined by specifying the input and bottleneck layers. This model is

used to extract the compressed representation of the input data.

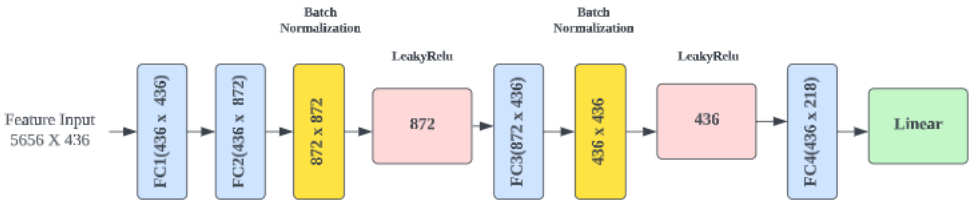

Fig 3 Autoencoder Model Architecture With Compression

The model architecture in Figure 3 focuses on a specific autoencoder architecture for compression. It uses a preprocessed speech dataset instead of a synthetic dataset.

The encoder and decoder architecture in this architecture consists of two dense layers each, with leaky ReLU activation and batch normalization. The bottleneck layer has a size that is half of the input features. The model is compiled with the Adam optimizer, mean squared error loss function and accuracy as a metric. The purpose of this autoencoder is to learn a compressed representation of the speech data, capturing important characteristics such as pitch, tempo, spectral frequencies, and rhythm. It also includes scaling the data using Min Max Scaler. In summary, both architectures implement autoencoder models, but they have different architectural designs and are applied to different datasets

5. Fusion of autoencoder with various machine learning approaches

The model constructed using autoencoders is used for constructing various accented speech models using linear regression, decision tree, support vector machine(SVM), random forest classifier(RFC), K nearest neighbor (KNN), stochastic gradient descent(SGD) and multilayer perceptron approaches.

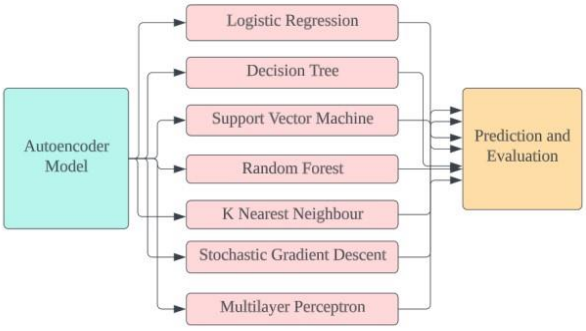

Fig 4 The Autoencode Machine Learning Fusion

6. Evaluation and Performance Metrics

We have evaluated the performance of the proposed approach using accuracy rates, word error rates(WER), and feature interpretability to assess the effectiveness of the combined self-supervised learning and autoencoder fusion method.

The performance evaluation in terms of accuracy, WER, and log loss generated in each experiment is shown in Table 1. Figure 1 shows the different experimental evaluations in terms of accuracy, loss, precision, and recall generated in different experiments.

Table 1 Performance Evaluation

Sl. No.	Methodology	Accuracy	WER	Log Loss
1.	Encoder Without Compression	66.86	34.24	0.00059833
2	Encoder With Compression	80.61	19.29	0.00072474
	Machine Learning Models with original data as Input			
3	Logistic Regression	89.39	19.69	0.558
4	Decision Tree	27.75	52.85	0.689
5	SVM	44.23	36.44	1.275
6	Random Forest	46.50	35.92	0.015
7	KNN	43	16.59	0.052

8	SGD	20.75	76.86	0.568
9	MLP	99.25	0.95	0.135
	Machine Learning Models with Input as Autoencoder model with compression			
10	Logistic Regression	94.54	15.35	0.03
11	Decision Tree	85.15	32.75	1.458
12	SVM	95.15	12.67	1.372
13	Random Forest	93.93	14.43	0.030
14	KNN	93.63	15.63	0.446
15	SGD	90.90	22.25	0.003
16	MLP	96.06	3.25	0.115

The machine learning models when trained with autoencoder data showed high improvement in performance.

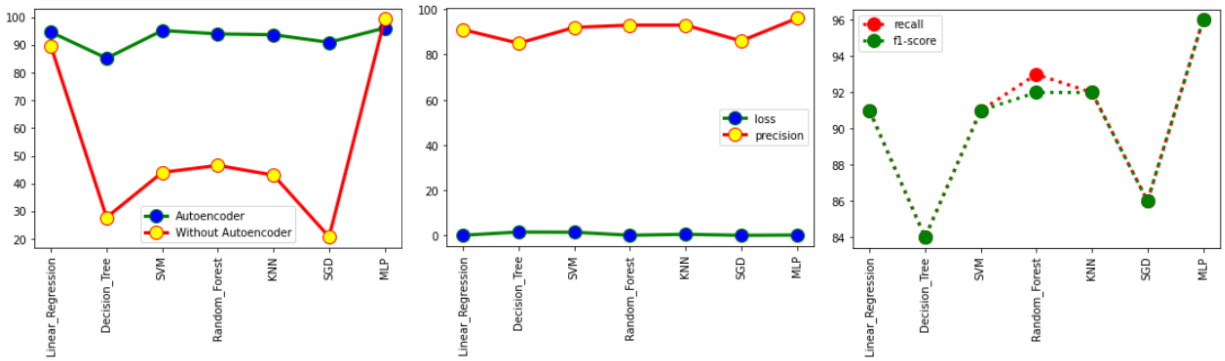

Fig 5 Performance Evaluation in terms of accuracy, loss, precision, recall, and f1-score

7. Comparative Analysis:

Finally, we compared the results of the proposed approach with existing methods for accent speech analysis in Malayalam and evaluated the advantages in terms of accuracy, robustness, and generalizability. Encoder-based models, particularly those with compression, outperform Machine Learning models with original data as input in terms of accuracy and word error rate (WER).

The Encoder With Compression approach achieved an accuracy of 80.61% and a reduced WER of 19.29%, indicating the effectiveness of incorporating compression techniques to improve performance.Among the Machine Learning models with original data as input, Logistic Regression shows the highest accuracy at 89.39% and a WER of 19.69%. However, other models such as Decision Tree, SVM, and SGD exhibit relatively poorer performance. Utilizing the compressed representations generated by the Autoencoder as input to Machine Learning models significantly enhances their performance. Logistic Regression, SVM, Random Forest, KNN, SGD, and MLP demonstrate notable improvements in accuracy and reduced WER when compared to the original data as input.

MLP consistently performs well in both scenarios, showcasing high accuracy and low WER. This suggests that MLP models are effective for speech recognition tasks, regardless of the input type. The experimental result implies that the Encoder With Compression approach combined with Machine Learning models, particularly Logistic Regression and SVM, yields the best overall performance in terms of accuracy and WER.

The comparative analysis highlights the effectiveness of encoder-based models, specifically those with compression, in improving accented speech recognition performance. Additionally, incorporating compressed representations from the Autoencoder as input to Machine Learning models enhances their accuracy and reduces word error rates. These findings can guide the selection of suitable methodologies for speech recognition tasks, with a focus on leveraging the advantages of compression techniques and appropriate model choices.

Table 2 Comparison with Existing Research

Reference	Methodology	Year	Accuracy	WER	Precision
[1]	Adversarial Autoencoders	2022	58.38	-	-
[2]	Discriminative Autoencoders	2022	-	5.10	-
[4]	Autoencoders	2019	-	18.0	-
[9]	Autoencoders	2021	78.67	-	-
[10]	Kaldi	2023	93.96	-	-
[11]	Mixed transform	2023	(87-93)	-	-
[12]	ANN and KNN	2023		-	91.26 & 91.5(for 2 different databases)

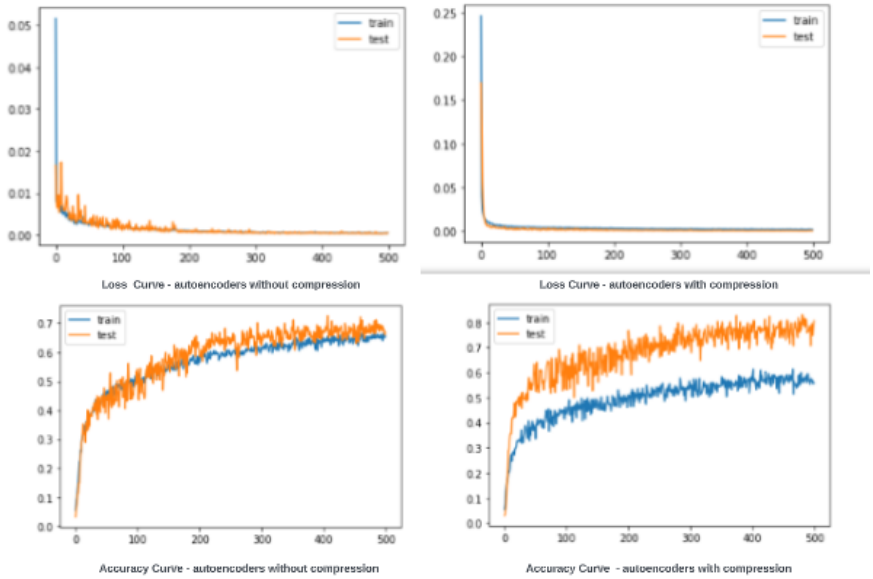

Fig 6 Learning Curves for autoencoder-based accent modeling

4. Conclusion and Future Work

In this research, we have investigated the use of autoencoders for accented speech recognition for Malayalam. Through our experiments and analysis, we have observed promising results indicating the effectiveness of autoencoders in capturing relevant features and extracting meaningful representations from speech data. The performance evaluation of our proposed approach demonstrated improved accuracy and robustness in accent recognition tasks. These techniques leverage the power of deep learning and unsupervised learning to improve the quality of extracted features and enhance the discriminative capabilities of the models.

While our research has yielded promising results, there are several avenues for further investigation and improvement.

The lack of a benchmark dataset for research in this area creates a significant research gap and poses challenges for conducting studies. The authors plan to address this gap by initiating the construction of an accented dataset specifically for Malayalam. The dataset will be made publicly available, enabling researchers to conduct various studies and advancements in the field of accented speech recognition. In the future, the authors aim to propose improved approaches for constructing unified accented models that can recognize all accents present in the Malayalam language. The research and methodologies developed for Malayalam can also be adopted and applied

to other low-resourced languages, further contributing to advancements in accented speech recognition.

References

[1] Sahu, S., Gupta, R., Sivaraman, G., AbdAlmageed, W., & Espy-Wilson, C. Y. (2017). Adversarial Auto-Encoders for Speech Based Emotion Recognition. https://doi.org/10.21437/interspeech.2017-1421

[2] Lee, H., Huang, P., Cheng, Y., & Wang, H. (2022). Chain-based Discriminative Autoencoders for Speech Recognition. arXiv (Cornell University). https://doi.org/10.48550/arxiv.2203.13687

[3] Deng, J., Xu, X., Zhang, Z., Frühholz, S., & Schuller, B. (2018). Semi supervised Autoencoders for Speech Emotion Recognition. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 26(1), 31–43. https://doi.org/10.1109/taslp.2017.2759338

[4] Karita, S., Watanabe, S., Iwata, T., Delcroix, M., Ogawa, A., & Nakatani, T. (2019). Semisupervised End-to-end Speech Recognition Using Text-to-speech and Autoencoders. https://doi.org/10.1109/icassp.2019.8682890

[5] Huang, P., Xu, H., Li, J., Baevski, A., Auli, M., Galuba, W., Metze, F., & Feichtenhofer, C. (2022). Masked Autoencoders that Listen. arXiv (Cornell University). https://doi.org/10.48550/arxiv.2207.06405

[6] Atmaja, B. T., & Sasou, A. (2022). Evaluating Self-Supervised Speech Representations for Speech Emotion Recognition. IEEE Access, 10, 124396–124407. https://doi.org/10.1109/access.2022.3225198

[7] Peng, S., Kai, C., Tian, T., & Jingying, C. (2022). An autoencoder-based feature level fusion for speech emotion recognition. Digital Communications and Networks. https://doi.org/10.1016/j.dcan.2022.10.018

[8] Bastanfard, A., & Abbasian, A. (2023). Speech emotion recognition in Persian based on stacked autoencoder by comparing local and global features. Multimedia Tools and Applications. https://doi.org/10.1007/s11042-023-15132-3

[9] Ying, Y., Tu, Y., & Zhou, H. (2021). Unsupervised Feature Learning for Speech Emotion Recognition Based on Autoencoder. Electronics, 10(17), 2086. https://doi.org/10.3390/electronics10172086

[10] Barkani, F., Hamidi, M., Laaidi, N. *et al.* Amazigh speech recognition based on the Kaldi ASR toolkit. *Int. j. inf. tecnol.* (2023). https://doi.org/10.1007/s41870-023-01354-z

[11] Abou-Loukh, S. J. . and Abdul-Razzaq, S. M. . (2023) "Isolated Word Speech Recognition Using Mixed Transform", *Journal of Engineering*, 19(10), pp. 1271–1286. doi: 10.31026/j.eng.2013.10.06.

[12] Al Dujaili, M.J., Ebrahimi-Moghadam, A. Automatic speech emotion recognition based on hybrid features with ANN, LDA and K_NN classifiers. *Multimed Tools Appl* (2023). https://doi.org/10.1007/s11042-023-15413-x