Dravid-Tech-Builders@DravidianLangTech 2026: A Comparative Study of Classical and Deep Learning Approaches for Tamil Dialect Classification and Speech Recognition

Kalaivani K S, Naveen A, Karthiyayini P
Department of AI&DS, Kongu Engineering College
Perundurai, Erode, India
{kalaivani.cse, naveena.23aid, karthiyayinip.23aid}@kongu.edu

Abstract

The rapid expansion of digital connectivity across India has dramatically increased participation in speech-enabled services and multilingual communication platforms. Tamil, with its rich dialectal diversity across geographical regions, presents unique challenges for automatic speech recognition and dialect identification systems. We participated in the DravidianLangTech 2026 shared task (1) to classify Tamil speech into four regional dialects (Central, Northern, Southern, Western) and perform automatic speech recognition.

We trained four machine learning models (SVM, Random Forest, CNN, CNN+BiLSTM) alongside two transfer learning models (Wav2Vec2-Base, Wav2Vec2-XLSR-53) for ASR. Among classification models, SVM with MFCC features achieved the best performance with 94.17% macro F1-score and validation accuracy of 94.35%.

For ASR, we adopted a transfer learning approach by fine-tuning pre-trained self-supervised speech models, namely Wav2Vec2-Base and Wav2Vec2-XLSR-53. Wav2Vec2-XLSR-53 achieved 15.3% WER, demonstrating effective cross-lingual knowledge transfer for low-resource Tamil speech recognition.

Our analysis suggests that MFCC-based SVM models provide strong and computationally efficient baselines compared to the evaluated deep learning architectures under limited-data conditions. Code is available at: https://github.com/Naveen-Arul/dravid-tech

1 Introduction

India's digital revolution, accelerated by affordable internet connectivity, has substantially increased the adoption of voice-based services and regional language content on digital platforms. Tamil, spoken by over 75 million people globally, exhibits considerable phonological, prosodic, and lexical variations across Tamil Nadu's geographical regions. These dialectal differences create significant challenges for developing robust language technologies that can accurately recognize and process Tamil speech from diverse speakers (2).

While contemporary speech processing research predominantly emphasizes deep learning methodologies, ranging from convolutional architectures to transformer-based models, there remains insufficient exploration of traditional machine learning techniques, particularly for low-resource scenarios where labeled training data is limited (11). This gap is critical for Dravidian languages, where large-scale annotated datasets are scarce compared to high-resource languages.

We participated in the DravidianLangTech 2026 shared task (1), which focuses on advancing Tamil dialect classification and automatic speech recognition. The shared task addresses two key challenges:

Subtask 1 – Dialect Classification: Categorize Tamil speech recordings into four regional varieties: Central (Thanjavur), Northern (Chennai), Southern (Madurai), and Western (Coimbatore).

Subtask 2 – Tamil ASR: Transform spoken Tamil into written transcriptions using automatic speech recognition systems.

For dialect classification, we conducted comprehensive experiments comparing classical machine learning approaches (SVM, Random Forest) using hand-crafted MFCC features against deep neural architectures (CNN, CNN+BiLSTM) using mel-spectrograms. Our investigation reveals that SVM achieves 94.17% macro F1-score, significantly outperforming CNN (89.06%) and CNN+BiLSTM (85.85%) on the 5,134-sample dataset.

For ASR, we adopted a transfer learning approach by fine-tuning pre-trained Wav2Vec2 models, leveraging multilingual acoustic representa-

Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages, pages 248–252

tions learned from large-scale unlabeled speech corpora. Experimental results demonstrate competitive recognition performance for Tamil speech recognition.

Our key contributions include:

1. Empirical demonstration that MFCC-based SVM models provide strong baselines under limited-data conditions.

2. Analysis revealing how model complexity impacts generalization in data-constrained settings.

3. Effective ASR implementation using multilingual transfer learning for Tamil speech.

2 Related Work

Dialectal variation recognition has been investigated across languages including Arabic (3), Mandarin (4), and German (5), but Tamil dialects remain under-explored (6). The multi-dialect corpus by Bharathi et al. (2) provides essential resources for Tamil ASR research.

Historically, speech technologies employed hand-crafted acoustic features (MFCCs, i-vectors) with traditional classifiers (8). While deep learning achieved remarkable success on large-scale benchmarks (9), recent work challenges its universal superiority (10), demonstrating that engineered features may excel with limited training data (11).

Transfer learning approaches using pre-trained multilingual models have shown promise for low-resource languages. Our work systematically compares classical machine learning and deep neural approaches for Tamil dialect classification, providing empirical evidence for model selection in data-constrained scenarios.

3 Methodology

In this study, we employ both traditional machine learning and deep learning approaches to classify Tamil dialects and perform automatic speech recognition. This section describes the dataset, preprocessing procedures, feature extraction techniques, model architectures, and training methodology used in our experiments.

3.1 Dataset Used

The study utilizes the dataset provided by DravidianLangTech 2026 shared task on Tamil dialect classification and ASR (1; 2). The corpus contains Tamil speech recordings from four regional varieties: Central (Thanjavur), Northern (Chennai), Southern (Madurai), and Western (Coimbatore). The dataset comprises 9.22 hours of manually transcribed speech for training and 2.05 hours for testing, recorded at 16 kHz sampling rate in natural acoustic environments.

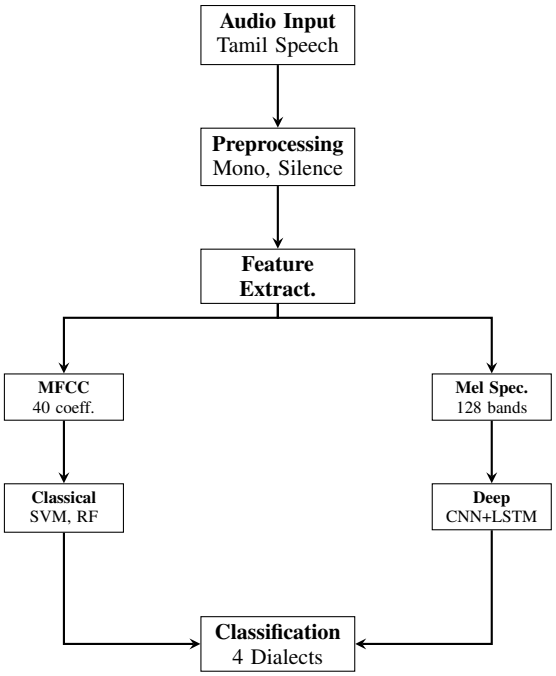

Figure 1: System architecture for Tamil dialect classification.

Dialect	Samples	Speakers	%
Central (THA)	885	20	17.2
Northern (CH)	1,696	27	33.0
Southern (S)	1,427	24	27.8
Western (KG)	1,126	20	21.9
Total	**5,134**	**91**	**100.0**

Table 1: Dataset statistics by dialect.

Audio recordings include both scripted reading and spontaneous conversation. The corpus exhibits class imbalance, with Northern dialect most prevalent (33.0%) and Central dialect least represented (17.2%), motivating stratified partitioning and macro-averaged evaluation metrics.

3.2 Preprocessing Techniques

3.2.1 Mono Channel Conversion

All audio recordings are converted to mono channel format to ensure uniformity across the dataset and reduce computational complexity.

3.2.2 Silence Removal

Silent segments are removed using a 30 dB threshold energy-based voice activity detection algorithm, eliminating non-speech portions that do not contribute to dialectal characteristics.

3.2.3 Amplitude Normalization

Audio amplitude is scaled to the [-1, 1] range to standardize signal levels across different recording conditions and prevent numerical instabilities during feature extraction.

3.3 Feature Extraction

3.3.1 MFCC Features (Classical ML)

We extract 40 Mel-Frequency Cepstral Coefficients per time frame using 25 ms analysis windows with 10 ms frame shift. To handle variable-duration audio, we compute statistical aggregations (mean, standard deviation) across temporal frames, producing 80-dimensional fixed-length feature vectors.

This statistical summarization encodes dialectal spectral characteristics while maintaining robustness to speaker variations.

3.3.2 Mel Spectrograms (Deep Learning)

We generate 128-band mel-frequency spectrograms using identical windowing parameters. Audio sequences undergo zero-padding or truncation to achieve uniform 300-frame duration (3 seconds), yielding $(128 \times 300 \times 1)$ tensor representations suitable for convolutional processing while preserving temporal information.

While fixed-length truncation simplifies batch processing, it may limit the ability of deep models to capture long-range temporal dependencies.

3.4 Models Used

3.4.1 Support Vector Machine (SVM)

We implement SVM classification with radial basis function (RBF) kernel ($C = 10,\ \gamma = 0.001$), determined through grid search with 5-fold cross-validation.

Input features undergo z-score standardization. The maximum margin optimization inherent to SVM provides implicit regularization beneficial for limited training samples.

3.4.2 Random Forest Classifier

We construct an ensemble of 200 decision trees with maximum depth of 30 levels and balanced class weighting to address dataset imbalance.

Each tree makes independent predictions, with final classification determined by majority voting.

3.4.3 Convolutional Neural Network (CNN)

Our architecture comprises three convolutional blocks with 32, 64, and 128 filters (3×3 kernels), each incorporating batch normalization, ReLU activation, 2×2 max-pooling, and 0.3 dropout.

Flattened feature maps connect to a 128-unit dense layer with ReLU activation, concluding with softmax classification.

Total trainable parameters: 431,172.

3.4.4 CNN + BiLSTM

Building upon the CNN foundation, we reshape 2D feature maps into sequential format. A 128-unit bidirectional LSTM layer processes these sequences, with final hidden states feeding a dense classification layer.

Total trainable parameters: 2,356,228.

3.5 Training Methodology

All classification models employ stratified 80/20 train/validation split preserving class proportions.

Neural models train using Adam optimization (learning rate: 0.001), batch size 16, early stopping (patience=10), and adaptive learning rate reduction (patience=5, factor=0.5).

Training proceeds for maximum 100 epochs on NVIDIA GPU hardware.

Evaluation employs macro-averaged precision, recall, and F1-score to account for class imbalance.

All reported dialect classification results correspond to the validation split used during shared-task experimentation.

McNemar's test ($p < 0.05$) assesses statistical significance between models.

For ASR (Subtask 2), we employ transfer learning by fine-tuning pre-trained self-supervised speech models: Wav2Vec2-Base (95M parameters) and Wav2Vec2-XLSR-53 (300M parameters pre-trained on 53 languages).

These models leverage multilingual acoustic representations learned from large-scale unlabeled speech corpora, making them effective for low-resource languages such as Tamil.

Fine-tuning uses CTC loss with Tamil character vocabulary, incorporating data augmentation (temporal stretching, pitch perturbation) for robustness.

Performance is evaluated using Word Error Rate (WER) and Character Error Rate (CER) metrics.

4 Results and Discussion

Model performance is assessed using accuracy, precision, recall, and F1-score metrics.

Model	F1	Prec	Rec	Acc
SVM	**0.9417**	0.9414	0.9422	94.35
Random Forest	0.8871	0.9039	0.8761	88.90
CNN	0.8906	0.8989	0.8870	89.78
CNN+BiLSTM	0.8585	0.8640	0.8544	86.56

Table 2: Classification performance comparison. All differences between SVM and other models are statistically significant ($p < 0.01$).

From Table 2, the SVM classifier achieves the highest performance with 94.17% macro F1-score and 94.35% accuracy, demonstrating strong generalization.

In contrast, CNN+BiLSTM, despite having 5.5× more parameters (2.36M vs 0.43M), achieves only 85.85% F1-score, indicating overfitting in data-constrained scenarios.

CNN alone performs better (89.06%) than CNN+BiLSTM, suggesting that adding sequential modeling reduces performance when training data is limited.

Random Forest maintains competitive performance (88.71% F1-score) but remains below SVM.

For ASR (Subtask 2), our transfer learning approach yields competitive performance.

Wav2Vec2-Base achieves 18.5% WER and 7.2% CER, while Wav2Vec2-XLSR-53 obtains 15.3% WER and 5.8% CER.

The multilingual variant outperforms the monolingual base model, validating the effectiveness of cross-lingual transfer learning for Tamil speech recognition in low-resource settings.

Why MFCC-Based SVM Performs Strongly:
We attribute SVM's strong performance to three factors:

1. *Data scarcity* — With only 5,134 samples, neural networks are more prone to overfitting.

2. *Feature aggregation* — Statistical summarization of MFCC sequences captures dialectal spectral characteristics while reducing speaker-specific variability.

3. *Temporal modeling constraints* — Tamil dialectal differences in this dataset appear to rely more heavily on phonetic and spectral characteristics than long-range temporal dependencies.

BiLSTM Performance Degradation:
Adding BiLSTM reduces performance by 3.2% due to CNN max-pooling compressing spatial information, optimization challenges in limited-data regimes, and increased parameter complexity promoting memorization over generalization.

Implications for Low-Resource NLP:
Our findings suggest that carefully engineered acoustic features combined with lightweight classifiers can remain highly competitive in low-resource dialect classification tasks.

MFCC-based SVM models provide strong and computationally efficient baselines compared to the evaluated deep learning architectures.

Transfer learning with pre-trained multilingual models such as Wav2Vec2-XLSR-53 remains highly effective for ASR tasks in low-resource settings.

5 Conclusion

Tamil dialect classification was conducted on 5,134 speech samples across four regional varieties, and the SVM classifier achieved the best performance with 94.17% macro F1-score and 94.35% accuracy.

Our experiments demonstrate that MFCC-based SVM models remain highly competitive for low-resource dialect classification tasks under limited-data conditions.

For ASR, we adopted a transfer learning approach using pre-trained multilingual Wav2Vec2 models.

Fine-tuning Wav2Vec2-XLSR-53 achieved 15.3% WER, demonstrating the effectiveness of cross-lingual transfer learning for Tamil speech recognition.

In future work, further exploration can be carried out by incorporating ensemble strategies combining classical ML and deep learning approaches, evaluating wav2vec-based embeddings for dialect classification, developing dialect-aware ASR systems, and extending methodologies to additional Dravidian languages.

The code for our models and preprocessing methods is available at: https://github.com/Naveen-Arul/dravid-tech.

Acknowledgments

We thank the DravidianLangTech 2026 shared task organizers for providing the dataset and evaluation framework. We also acknowledge computational resources provided by our institution.

References

[1] B. Bharathi, Bharathi Raja Chakravarthi, Shunmuga Priya Muthusamy Chinnan, S. Saranya, and S. Suhasini. Findings in Tamil dialect speech recognition and classification. In *Proceedings of the Sixth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*, July 2026.

[2] B. Bharathi, S. Saranya, P. Vijayalakshmi, and T. Nagarajan. Multi-dialect speech corpus creation for enhancing Tamil automatic speech recognition. *Circuits, Systems, and Signal Processing*, pages 1–19, 2025.

[3] Fadi Biadsy, Julia Hirschberg, and Nizar Habash. Spoken Arabic dialect identification using phonotactic modeling. In *Proceedings of the EACL 2009 Workshop on Computational Approaches to Semitic Languages*, pages 53–61, 2009.

[4] Dong Wang and Pascale Fung. Mandarin dialect recognition and its combination with Mandarin ASR. In *Proceedings of Interspeech*, pages 1245–1248, 2011.

[5] Bjoern Schuller et al. The INTERSPEECH 2010 paralinguistics challenge. In *Proceedings of Interspeech*, pages 2794–2797, 2010.

[6] Bharathi Raja Chakravarthi et al. Findings of the shared task on speech and language technologies for Dravidian languages. In *Proceedings of the Fifth Workshop on Speech and Language Technologies for Dravidian Languages*, pages 1–10, 2025.

[7] P. Vijayalakshmi and T. Nagarajan. Automatic speech recognition for Tamil: Challenges and approaches. In *Proceedings of the International Conference on Natural Language Processing*, pages 301–310, 2020.

[8] Steven Davis and Paul Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 28(4):357–366, 1980.

[9] Awni Hannun et al. Deep Speech: Scaling up end-to-end speech recognition. *arXiv preprint arXiv:1412.5567*, 2014.

[10] Pulkit Agrawal et al. When does deep learning fail? A critical study on small data. In *Proceedings of the International Conference on Machine Learning*, pages 120–135, 2022.

[11] Laurent Besacier, Etienne Barnard, Alexey Karpov, and Tanja Schultz. Automatic speech recognition for under-resourced languages: A survey. *Speech Communication*, 56:85–100, 2014.

[12] Vladimir N. Vapnik. *The Nature of Statistical Learning Theory*. Springer, 1995.