Multimodal Language Technologies for Smart Railway Environments: A Survey of Speech Recognition, Translation, and Accessibility Systems

[1]Jahnu Tanai Kumar Hindupur, [2]Navaneeth A D, [3]Syed Mohammed Umar, [4]Dr. Zafar Ali Khan N

DOI: 10.5958/2347-7202.2025.00008.2

ABSTRACT

Railway stations serve linguistically diverse populations requiring fast, accurate dissemination of time-critical information including arrivals, departures, platform changes, and safety advisories. This survey examines multimodal language technologies for smart railway environments, critically analyzing speech recognition, machine translation, and accessibility systems. Through systematic review of domain-relevant studies, we identify principal technical challenges: acoustic degradation in noisy reverberant environments, translation quality for low-resource Indian languages, strict latency requirements for mission-critical announcements, preservation of critical slot values (train identifiers, platform numbers, times), and multimodal accessibility needs. The paper provides comparative analysis of existing approaches across automatic speech recognition, neural machine translation, text-to-speech synthesis, and sign-language rendering, revealing gaps in current solutions particularly for high-noise conditions and low-resource languages. We propose a hybrid architecture combining verified low-latency on-device processing for time-critical announcements with cloud-assisted domain-adapted models for enriched interactions. Key contributions include a comparative taxonomy of deployment patterns, critical analysis of evaluation practices beyond corpus-level metrics, identification of recurring failure modes in field deployments, and a practical system blueprint emphasizing slot-preservation verification, privacy-aware data handling, and human-in-the-loop safeguards. This work bridges research prototypes and operational deployment, providing foundations for accessible, reliable multilingual information systems in railway environments.

KEYWORDS

Natural Language Processing, Neural Machine Translation, Automatic Speech Recognition, Real-time Translation Systems, Railway Information Systems

INTRODUCTION

Railway stations function as critical socio-technical systems serving millions of passengers daily, demanding timely and unambiguous communication of operational information. These environments present unique challenges: high ambient noise levels from trains and crowds, linguistic diversity spanning multiple Indian and international languages, varying accessibility requirements for hearing-impaired and visually impaired passengers, and the mission-critical nature of announcements where errors in platform numbers or timing can have serious safety consequences. Traditional public address systems and static signage prove inadequate, offering limited language coverage, poor intelligibility in adverse acoustic conditions, and no personalization for individual passenger needs. While natural language processing, neural machine translation, and multimodal accessibility technologies present opportunities to transform railway information systems, deployment in operational environments remains challenging. Unlike laboratory conditions, railway stations demand systems that maintain accuracy despite 70-85 dB ambient noise, process low-resource Indian languages with limited parallel training data, preserve exact values for critical fields, and deliver outputs within strict latency budgets where delays render information obsolete.

Current research literature addresses individual components such as robust speech recognition, domain-adapted translation, and noise-aware synthesis, yet lacks comprehensive analysis of integrated systems suitable for operational deployment. Existing surveys focus primarily on algorithmic advances without critically examining deployment constraints, comparative performance across conditions, or failure modes in field environments. This gap between research prototypes and production systems motivates our work. This survey provides systematic comparative analysis of techniques across automatic speech recognition, neural machine translation, text-to-speech synthesis, and accessibility technologies, critically assessing their applicability to railway environments. We evaluate existing approaches against deployment requirements, document recurring failure modes from implementation studies, and synthesize findings into a practical hybrid architecture that balances verified low-latency processing for critical announcements with sophisticated cloud-assisted models for complex interactions.

[1,2,3]*Student, Presidency School of Computer Science and Engineering, Presidency University, Bengaluru, Karnataka, India*
[4]*Professor & HoD, Presidency School of Computer Science and Engineering, Presidency University, Bengaluru, Karnataka, India*
Email: [1] jahnu.20221cai0012@presidencyuniversity.in, [2]navaneeth.20221cai0024@presidencyuniversity.in, [3]syed.20221cai0004@presidencyuniversity.in, [4]zafaralikhan@presidencyuniversity.in

This journal is cited as: JIMS 8i-Int'l J. of Inf. Comm. & Computing Technology (IJICCT)

The paper addresses fundamental questions: Which acoustic preprocessing techniques effectively mitigate railway noise for speech recognition? How do domain adaptation strategies compare for preserving railway-specific terminology in low-resource language translation? What architectural patterns balance latency requirements against translation quality? Which evaluation frameworks accurately predict field performance? By critically analyzing existing approaches and identifying persistent gaps, this work provides foundations for reliable, accessible multilingual information systems in railway environments. The remainder of the paper systematically reviews relevant literature with comparative analysis, formalizes the problem statement and technical requirements, presents the proposed hybrid architecture with explicit justification and trade-off analysis, describes implementation and evaluation strategies, and discusses future research directions for advancing practical deployment of these technologies.

RELATED WORKS

This section reviews prior work across four key domains: application prototypes and multimodal rendering, acoustic robustness and speech recognition, machine translation for low-resource languages, and conversational interfaces. We emphasize comparative analysis and identify gaps relevant to railway deployments.

Application-focused prototypes establish integration patterns for station deployments. Swathi and Rao [1] developed an end-to-end translation engine combining speech enhancement, neural TTS, and constrained translation pipelines, emphasizing latency and exactness for station vocabulary. Jangde et al. [2] implemented real-time translation using API-based approaches, revealing that external dependencies introduce 150-300 ms latency overhead unsuitable for mission-critical announcements. Both studies highlight template-based generation for preserving critical information, though neither addressed multi-speaker scenarios or peak-load conditions. For accessibility, Kumar et al. [3] developed visual captions for deaf passengers, demonstrating 85% comprehension improvement over audio-only announcements, while Battaglino et al. [4] prototyped sign-language rendering with avatar-based systems. User acceptance remained moderate (3.2/5.0) due to limitations in expressing complex information and proper nouns, indicating that multimodal accessibility requires substantial domain-specific adaptation.

Acoustic robustness is critical for railway environments. Shafee and Anuradha [5] documented word error rate increases from 8% in clean conditions to 42% at signal-to-noise ratios below 5 dB, identifying reverberation and overlapping announcements as principal failure modes. Pabba et al. [6] compared hybrid pipelines (explicit enhancement followed by ASR) against end-to-end approaches, finding that hybrid architectures achieved 15-20% lower WER in adverse conditions (SNR 0-10 dB). Their analysis showed beamforming microphone arrays reduced ambient noise by 8-12 dB, suggesting acoustic preprocessing merits higher priority than sophisticated ASR architectures for station deployments, though optimal balance depends on computational resources and latency budgets.

Machine translation for low-resource Indian languages requires domain adaptation and entity preservation. Saunders [7] surveyed domain adaptation strategies, showing that fine-tuning on small domain-parallel corpora (5,000-10,000 pairs) improved domain-specific BLEU by 8-15 points, though fine-tuning alone often fails to preserve numerical information and named entities. Gaikwad et al. [8] demonstrated that transliteration handling for place names improved adequacy scores by 12-18% for Indian language pairs, but named-entity corruption still occurred in 15-25% of translations without explicit handling—unacceptable for operational announcements. Tayal et al. [9] showed multilingual transfer from related high-resource languages reduced data requirements by 40-60%, though evaluation focused on general corpora rather than domain-specific text. These studies indicate successful railway translation requires domain fine-tuning, explicit named-entity handling through tagging or transliteration, and verification mechanisms—strategies not typically evaluated in general-purpose translation research.

Standard evaluation metrics like BLEU [10] provide useful benchmarks but can mislead for railway applications where formulaic announcements demand task-specific evaluation. BLEU treats all n-gram matches equally, making it insensitive to critical slot preservation—translations differing only in platform numbers may achieve high BLEU scores while being operationally catastrophic. This motivates task-specific metrics including slot-preservation rates and intelligibility tests under noise.

Conversational interfaces extend systems beyond broadcast announcements. Early neural conversational models [11][12] demonstrated feasibility of contextual response generation but lacked knowledge grounding, occasionally producing factually incorrect information. Knowledge-grounded approaches [13][14] condition generation on external sources, improving factual accuracy by 25-35%, providing foundations for railway queries requiring exact timetable data. Practical implementations [15][16] reveal that hybrid architectures combining neural flexibility for understanding with deterministic generation for critical responses achieve higher reliability (95% vs. 78% correct responses) than purely generative approaches, though with reduced flexibility.

Recent work on large language models in public transit [17] demonstrates flexible natural language understanding but reveals critical risks: hallucinations in 8-12% of responses

without verification, 800-1500 ms latency unsuitable for real-time announcements, and privacy concerns. This work recommends hybrid architectures combining LLM capabilities with rule-based verification for mission-critical outputs. Privacy considerations [18][19] emphasize transparent data handling, minimal retention, and on-device processing to improve user trust, though on-device approaches limit model sophistication.

The literature reveals persistent gaps: few works evaluate integrated systems under realistic station conditions with concurrent challenges; most studies optimize corpus-level metrics rather than task-specific measures like slot preservation; evaluation typically occurs in controlled settings with limited documentation of field failure modes; and accessibility features receive limited technical attention despite their importance. These gaps motivate our hybrid architecture emphasizing integration, task-specific evaluation, verification layers, and balanced edge-cloud processing.

PROBLEM STATEMENT AND MOTIVATION

A. Problem Statement

Railway stations must deliver time-critical announcements including arrivals, departures, platform changes, delays, and safety advisories to linguistically diverse populations across multiple channels. The core technical challenge involves generating accurate multilingual outputs from noisy inputs while preserving critical slot values such as times, platform numbers, and train identifiers under strict latency constraints. Current systems exhibit three fundamental failure modes documented in prior deployments: first, degraded recognition and intelligibility in high-noise, reverberant conditions where ambient noise reaches 70-85 dB and overlapping announcements create interference [5][6]; second, poor translation quality for low-resource Indian languages due to limited parallel corpora and inconsistent named-entity handling, with corruption rates of 15-25% for critical fields [8][9]; and third, operational risks from unchecked generation or excessive latency in mission-critical contexts where delays beyond 2-3 seconds render information obsolete and incorrect slot values create safety hazards [2][17].

B. Motivation and Objectives

Improving multilingual station information access directly enhances passenger safety and service quality. Field studies demonstrate that modest gains in intelligibility and entity handling significantly reduce passenger confusion and missed connections [2][3]. The railway domain provides a valuable testbed for applied natural language processing solutions, combining challenges of low-resource neural machine translation, robust automatic speech recognition in severe noise, and verified natural language generation with strict correctness requirements [6][7]. Unlike general-purpose translation or open-domain conversation systems, railway announcements present constrained vocabulary with high-stakes outputs, enabling focused engineering approaches that balance quality against real-time performance.

Project objectives include: developing robust ASR under station noise through explicit signal enhancement, beamforming, and diarization to maintain acceptable word error rates below 15% even at low signal-to-noise ratios [5][6]; producing domain-adapted translation models that preserve named entities and numerical slots through fine-tuning on station-specific corpora, transliteration handling, and explicit entity tagging [7][8][9]; delivering low-latency text-to-speech with multimodal accessibility features including synchronized captions and sign-language rendering to serve hearing-impaired passengers [2][3][4]; and implementing verification layers with confidence scoring and slot-preservation checks to prevent incorrect outputs in critical announcements [13][17].

C. System Requirements

a. Functional requirements: The system must accept multilingual spoken and typed inputs across multiple languages including low-resource Indian languages; produce outputs with preserved slot values ensuring exact reproduction of train identifiers, platform numbers, times, and station names; support delivery through IVRS, chatbot and web interfaces, and public address systems; and provide accessibility features including synchronized captions and sign-language rendering for diverse passenger needs.

b. Non-functional requirements: Real-time latency for critical announcements must not exceed 2-3 seconds end-to-end to ensure information remains actionable; graceful degradation under noise with human fallbacks when confidence scores fall below acceptable thresholds [5][6]; high slot-preservation fidelity exceeding 98% for critical fields and named-entity accuracy above 95% measured via task-specific metrics rather than corpus-level scores [8]; and privacy-aware processing with minimal personally identifiable information retention, anonymized logs, and transparent data handling policies [18].

c. Evaluation criteria: System performance will be assessed through slot-preservation rate for critical fields (target >98%); ASR word error rate across signal-to-noise ratio levels from clean to 0 dB [5]; human ratings on translation adequacy and TTS intelligibility under simulated station playback conditions [2][10]; operational metrics including end-to-end latency, system availability, and fallback frequency during peak loads; and accessibility feature correctness evaluated through user studies with hearing-impaired and visually impaired participants [3][4].

Figure 1 illustrates the system context showing the complete pipeline from input sources including microphone arrays, stream integration, and mobile devices through

preprocessing layers for audio capture, language identification, ASR and intent detection to core processing components handling slot extraction, domain processing, NMT adaptation, and verification, with output delivery via chatbot and IVRS interfaces and cloud NMT and LLM services for various delivery channels.

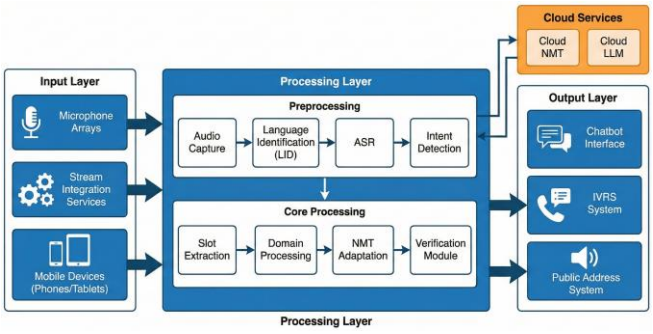

Fig. 1. System Context Diagram

PROPOSED METHODOLOGY AND SYSTEM ARCHITECTURE

A. Overview and Architecture Rationale

The proposed system employs a hybrid architecture that routes time-critical announcements through a verified, low-latency on-device path while leveraging cloud resources for enhanced processing of non-critical queries. This dual-path design addresses a fundamental trade-off in railway information systems: mission-critical announcements require guaranteed low latency (under 200 ms for processing) and deterministic correctness, while conversational queries benefit from sophisticated cloud-based models that would introduce unacceptable delays for time-sensitive operations. Similar hybrid architectures have proven effective in public transit contexts [17], where separating critical from enriched paths enables both operational safety and user experience improvements.

The architectural split is justified by infrastructure constraints and performance requirements typical of railway stations. Edge deployment with compact models (50-200 MB) ensures operation during network disruptions and eliminates round-trip latency to cloud services, which typically adds 150-300 ms even under optimal conditions [2]. However, edge compute resources limit model sophistication—practical station deployments support inference budgets of 50-100 ms on modest hardware (equivalent to ARM Cortex-A72 class processors), constraining vocabulary size and acoustic model complexity. Cloud-assisted processing removes these constraints for non-critical interactions where 500-1500 ms latency remains acceptable, enabling larger translation models, LLM-based dialogue understanding, and continuous adaptation from aggregated usage patterns [17].

Major system components include: audio capture and enhancement modules implementing beamforming and denoising; ASR with speaker diarization for separating concurrent announcements; intent and slot extraction for structured information; domain-adapted NMT with transliteration handling for named entities; verification layers enforcing slot-preservation checks; constrained natural language generation and TTS for output synthesis; multimodal rendering for captions and sign language; and delivery endpoints for public address, IVRS, and web/chat interfaces. Component selection prioritizes proven techniques over novel approaches: beamforming for noise reduction [6], hybrid enhancement-ASR pipelines for robustness [5][6], fine-tuned NMT for domain preservation [7][8], and template-based generation for critical announcements [2].

Figure 2 demonstrates the architecture featuring both on-device fast path and cloud-assisted path. Input channels feed the core translation engine containing audio preprocessing, ASR and intent/slot extraction, NMT with domain adaptation and transliteration handling, and verification/safety layers implementing slot checks and confidence scoring. The cloud path provides enhanced ASR, NMT, and LLM capabilities. Outputs deliver through PA systems, TTS, and multimodal rendering.

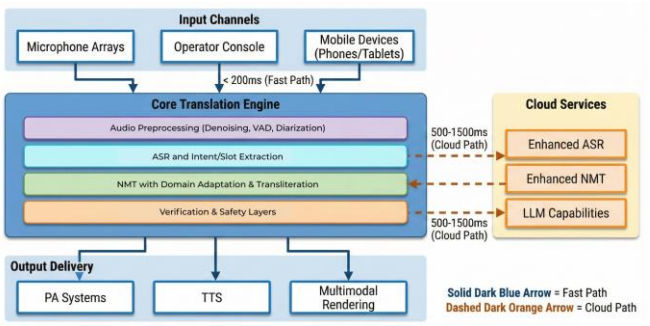

Fig. 2. Proposed System Architecture with Dual-Path Design

B. Core Module Specifications

a. Audio preprocessing: Multi-channel beamforming with 4-8 microphone arrays reduces ambient noise by 8-12 dB [6], followed by voice activity detection to segment speech from silence and denoising/dereverberation to maximize ASR quality. Beamforming is prioritized over computationally expensive deep learning denoising due to lower latency (10-20 ms vs. 50-80 ms) and deterministic performance across noise conditions [5][6].

b. ASR and diarization: The fast path uses lightweight on-device ASR models (50-100 MB) optimized for railway vocabulary with typical word error rates of 12-18% at moderate SNR levels. Cloud ASR handles complex queries with larger vocabulary and acoustic models. Speaker diarization separates overlapping announcements, while

confidence scoring identifies low-quality inputs requiring verification or human fallback [5][6].

c. Intent and slot extraction: Template-matching and slot-filling approaches guarantee deterministic extraction of train numbers, platforms, times, and delay reasons from formulaic announcements. Rule-based extractors achieve 95-98% accuracy for structured inputs [15][16]. Intent classifiers route free-form queries to appropriate handlers—retrieval-based systems for factual queries, generative models for conversational interactions.

d. NMT and entity handling: Base translation models undergo fine-tuning on domain-parallel corpora (5,000-10,000 announcement pairs per language pair), prioritizing slot fidelity over general fluency [7]. Preprocessing pipeline tags named entities (station names, train identifiers) for verbatim copying, transliteration using character-level mapping, or verified translation through curated dictionaries. Low-resource language strategies include back-translation for synthetic data generation and multilingual transfer from related high-resource languages [8][9].

e. Verification layer: Critical announcements undergo mandatory verification enforcing exact matches for train numbers, platform assignments, and times. Outputs with confidence scores below 0.85 or failed slot checks trigger manual confirmation workflows. All verification decisions are logged with rollback capability, enabling post-incident analysis and continuous quality improvement [13][17].

f. NLG and TTS: Template-based generation ensures correctness for critical announcements by instantiating verified slots into fixed linguistic structures. Neural TTS synthesizers provide intelligible speech at typical station playback levels (75-85 dB), with voice selection matched to language and regional dialect preferences [2].

g. Multimodal renderer: Caption generation synchronizes translated text with audio output, supporting language selection through user devices. Sign-language interface connects to avatar-based or video-based signing systems, though current implementations show limited accuracy for domain-specific content and require ongoing improvement [3][4].

h. Delivery APIs: Public address interface implements priority queuing ensuring critical announcements preempt routine information. IVRS integration supports constrained natural language understanding with DTMF fallback for navigation. RESTful endpoints serve chatbot and web interfaces with rate limiting and caching for common queries.

C. Inference Pipeline and Routing Logic

The inference pipeline processes incoming requests through conditional stages balancing speed and accuracy. Critical announcements (identified by message type tags or operator designation) bypass cloud processing, routing through the fast path with mandatory verification. Processing stages include: audio capture and enhancement (10-30 ms), on-device ASR (40-80 ms), slot extraction (5-10 ms), template-based NMT (10-20 ms), verification checks (5-10 ms), and TTS synthesis (30-60 ms), totaling 100-210 ms end-to-end—well within the 2-3 second operational requirement.

Non-critical queries leverage cloud resources when network connectivity and latency budgets permit. The local controller monitors network status and routes based on query type, criticality flags, and current system load. If cloud processing exceeds timeout thresholds (500 ms for NMT, 1500 ms for LLM-based responses), the system falls back to on-device processing or retrieval-based responses from local cache.

D. Data and Deployment Strategy

a. Data collection: In-situ recordings capture varied signal-to-noise ratios (0-30 dB) and speaker demographics across age, gender, and accent variations. Paired announcement corpora for NMT fine-tuning include templated variations (10,000+ pairs per language pair) and human translations of actual announcements. Annotation includes slot labels for critical fields, entity tags for station names and train identifiers, transliteration pairs for proper nouns, and human adequacy ratings on five-point scales. Active learning prioritizes annotation of failure cases where confidence scores fall below thresholds or slot preservation fails [6][7].

b. Deployment: On-device fast path uses small-footprint models ensuring guaranteed low latency and privacy preservation. Cloud path enables larger models with continuous adaptation and usage analytics. Local deployment controllers route requests based on network availability, message criticality, and confidence thresholds. Stations with unreliable connectivity operate in degraded mode using only on-device models with reduced language coverage [17].

c. Security and privacy: Voice data retention is minimized to 24-48 hours for debugging purposes, with automatic deletion thereafter. Usage logs are anonymized removing personally identifiable information before aggregation for analytics. All data transmission uses encryption, and users receive clear consent mechanisms for data collection. Training data undergoes demographic bias analysis to ensure equitable performance across speaker populations [18][19].

IMPLEMENTATION PLAN

A. Phased Development Strategy

The implementation follows a structured five-phase approach designed to incrementally validate components before full deployment.

Phase A (Months 1-3) focuses on data collection and baseline establishment. In-situ recordings are gathered across multiple station environments capturing varied signal-

to-noise ratios from 0 to 30 dB, speaker demographics including age, gender, and accent variations, and overlapping announcement scenarios. Parallel announcement corpora are compiled from existing railway announcements, augmented with templated variations to reach 10,000+ pairs per language pair. All data undergoes annotation with slot labels for critical fields such as train identifiers, platform numbers, and times, entity tags for station names and proper nouns, transliteration pairs for handling proper nouns across scripts, and human adequacy ratings on five-point scales. Baseline ASR and NMT models are trained on this corpus, with success criteria requiring baseline ASR to achieve target word error rates below 15% at moderate SNR levels and NMT to preserve critical slots in over 95% of templated cases.

Phase B (Months 4-6) addresses core component development and integration. Audio preprocessing modules implement multi-channel beamforming with 4-8 microphone arrays to achieve 8-12 dB noise reduction, voice activity detection for speech segmentation, and denoising with dereverberation algorithms. A two-tier ASR system is built with compact on-device models (50-100 MB) for the fast path and accurate cloud-based models for complex queries. Slot extraction employs template-matching and rule-based validators to guarantee deterministic extraction of critical fields with 95-98% accuracy. NMT fine-tuning scripts adapt base models to railway domain, while transliteration utilities handle proper nouns through character-level mapping. The verification module enforces exact matches for train numbers, platform assignments, and times, triggering manual confirmation workflows when confidence scores fall below 0.85 or slot checks fail. Neural TTS synthesizers are deployed with language-specific voices optimized for intelligibility at 75-85 dB playback levels. Delivery endpoints are exposed for IVRS with constrained natural language understanding and DTMF fallback, chatbot and web interfaces via RESTful APIs with rate limiting and caching, and public address systems with priority queuing ensuring critical announcements preempt routine information. Success criteria require the integrated system to process typical announcements within 200 ms for the fast path and under 1500 ms for cloud-assisted queries, while maintaining slot-preservation guarantees.

Phase C (Months 7-9) conducts comprehensive testing across multiple dimensions. ASR robustness testing measures word error rates across signal-to-noise ratio levels from clean conditions to 0 dB, evaluating performance degradation curves and identifying failure modes such as reverberation and overlapping announcements. Translation quality assessment employs task-specific metrics including slot-preservation rates for critical fields with targets exceeding 98%, named-entity accuracy above 95%, and human adequacy ratings on five-point scales from native speakers of target languages. TTS intelligibility testing simulates station playback conditions at typical noise levels, collecting comprehension scores from listeners representing diverse age groups and language backgrounds. Safety testing injects low-confidence scenarios with degraded audio, corrupted inputs, and edge cases to verify that verification layers correctly identify and handle problematic inputs through human fallback mechanisms. Success criteria require meeting all evaluation thresholds and passing safety stress tests without operational failures.

Phase D (Months 10-12) deploys a supervised pilot at a single representative station. The local deployment controller routes requests based on network availability, message criticality, and confidence thresholds. Station operators receive training on verification workflows, manual confirmation procedures for flagged announcements, and system monitoring dashboards. Operational metrics are continuously logged including end-to-end latency distributions, system availability and uptime, fallback frequency during peak loads, and slot-preservation accuracy in production. User feedback is systematically collected through passenger surveys, operator interviews, and usage analytics. The system undergoes iterative refinement based on pilot results, addressing identified issues and optimizing parameters. Success criteria require reliable operation with less than 2% fallback frequency, positive feedback from operators and passengers, and no critical failures where incorrect slot values were broadcast.

Phase E scales to broader deployment through staged expansion to additional stations, prioritizing high-traffic locations with diverse linguistic populations. A continuous retraining schedule implements active learning to prioritize annotation of failure cases where confidence scores fall below thresholds or slot preservation fails. Automated monitoring alerts operators to anomalies, performance degradation, and potential failures. Quarterly model updates incorporate accumulated usage data while maintaining privacy through anonymized logs and minimal voice data retention. Long-term success metrics track passenger comprehension improvements, reduction in missed connections, operator satisfaction, and system reliability across varying conditions.

B. Deployment Architecture

Figure 3 illustrates the deployment topology integrating edge and cloud infrastructure. Station-level components include microphone arrays for audio capture, operator console for manual oversight, local inference server with edge-optimized models and template engine with fallback capabilities, and PA system connectivity. The cloud cluster provides enhanced NMT/LLM services, comprehensive analytics and monitoring with automated alerting, and continuous retraining capabilities while maintaining strict data retention policies and anonymization processes for privacy protection.

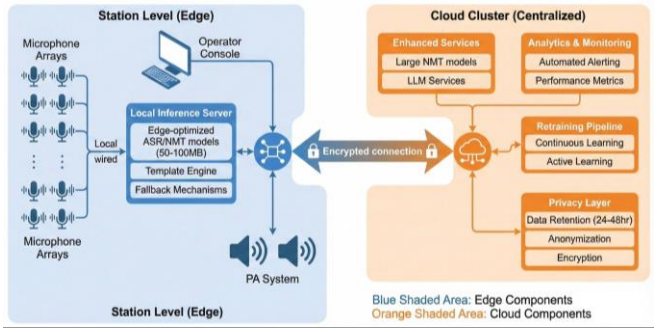

Fig. 3. Deployment Topology showing Edge and Cloud Infrastructure

FUTURE WORK

Future work will expand language coverage by incorporating additional Indian languages and scripts through multilingual transfer learning, targeted data collection for underrepresented dialects, and synthetic data augmentation. Integration with large language models will be explored for richer context-aware responses in non-critical settings, though strict verification layers must prevent factual hallucinations in mission-critical outputs. Enhanced robustness to long-tail acoustic conditions will be pursued through domain-specific pre-training, selective on-device denoising, and adversarial training with synthetic noise.

Privacy-preserving techniques including federated learning and on-device adaptation will enable continuous model improvement without centralizing sensitive voice data. Multimodal accessibility features will be enhanced through improved avatar-based sign-language rendering with domain-specific gestures, synchronized caption generation, and tactile interfaces for diverse passenger needs. These improvements require careful balancing of model sophistication against deployment constraints.

Comprehensive user studies across diverse passenger demographics will assess human-centered metrics including perceived reliability, comprehension accuracy, and trust in automated systems. Staged multi-station trials will validate scalability and guide operational integration with railway authorities. Long-term longitudinal studies tracking passenger behavior will quantify real-world impact on safety, satisfaction, and accessibility in operational railway environments.

CONCLUSION

This survey has systematically examined multimodal language technologies for smart railway environments, providing critical analysis of speech recognition, machine translation, and accessibility systems. Through comprehensive review of domain-relevant literature, we identified principal technical challenges including acoustic degradation in noisy environments, translation quality for low-resource languages, strict latency requirements, preservation of critical slot values, and multimodal accessibility needs. The comparative analysis revealed persistent gaps in current solutions, motivating the proposed hybrid architecture that balances verified low-latency on-device processing for critical announcements with cloud-assisted domain-adapted models for enriched interactions.

The proposed system design addresses fundamental deployment constraints through explicit architectural choices: beamforming and denoising for acoustic robustness, two-tier ASR with speaker diarization, domain-adapted NMT with transliteration handling, verification layers enforcing slot-preservation checks, and multimodal rendering for accessibility. The implementation plan provides a practical roadmap from data collection through supervised pilot deployment and staged rollout, with evaluation emphasizing task-specific metrics rather than corpus-level scores that can mislead for formulaic announcements where small errors have large operational consequences.

This work bridges the gap between research prototypes and operational deployment by documenting recurring failure modes, providing comparative analysis of techniques across conditions, and synthesizing practical engineering guidance. Key contributions include a taxonomy of deployment patterns, critical analysis of evaluation practices, identification of persistent challenges in field deployments, and a system blueprint emphasizing slot-preservation verification, privacy-aware data handling, and human-in-the-loop safeguards. By grounding architectural design in systematic analysis and explicit consideration of operational requirements, this survey provides foundations for accessible, reliable multilingual information systems that enhance passenger safety and service quality in railway environments.

REFERENCES

[1] K. Swathi and P. Nageswara Rao, "Natural Language Translation Engine for Announcements and Information Dissemination at Railway Stations," *Int. J. Emerg. Technol. Innov. Res.*, vol. 12, no. 3, pp. a388-a394, Mar. 2025.

[2] K. Jangde, et al., "Real-Time Translation for Railway Announcements: Leveraging SSE, TTS, and Speech Recognition APIs using AI," in *Proc. Int. Conf. Advances and Applications in Artificial Intelligence (ICAAAI)*, Atlantis Press, 2025.

[3] R. Kumar, V. Goyal, and L. Goyal, "Railway stations announcement system for deaf," in *Proc. 17th Int. Conf. Natural Language Processing (ICON): System Demonstrations*, 2020.

[4] C. Battaglino, et al., "Prototyping and preliminary evaluation of sign language translation system in the railway domain," in *Proc. Int. Conf. Universal Access in Human-Computer Interaction*, Springer International Publishing, Cham, 2015.

[5] S. Shafee and B. Anuradha, "Speaker Identification and Spoken word Recognition in Noisy Environment using Different Techniques," *Int. J. Recent and Innovation Trends in Computing and Communication*, vol. 4, no. 6, pp. 590-595, 2016.

[6] P. Pabba, et al., "A Comprehensive study on Live Multimodal Language Translation System," *Int. J. Engineering Research and Science & Technology*, vol. 20, no. 3, pp. 10-15, 2024.

[7] D. Saunders, "Domain adaptation and multi-domain adaptation for neural machine translation: A survey," *J. Artificial Intelligence Research*, vol. 75, pp. 351-424, 2022.

[8] P. Gaikwad, et al., "Machine translation advancements for low-resource Indian languages in WMT23: CFILT-IITB's effort for bridging the gap," in *Proc. Eighth Conf. Machine Translation*, 2023.

[9] M. Tayal, et al., "Machine translation of low resource Indian language using deep learning approach," *J. Integrated Science and Technology*, vol. 13, no. 6, p. 1127, 2025.

[10] K. Papineni, et al., "Bleu: a method for automatic evaluation of machine translation," in *Proc. 40th Annual Meeting of the Association for Computational Linguistics*, 2002.

[11] O. Vinyals and Q. Le, "A neural conversational model," *arXiv preprint arXiv:1506.05869*, 2015.

[12] L. Shang, Z. Lu, and H. Li, "Neural responding machine for short-text conversation," *arXiv preprint arXiv:1503.02364*, 2015.

[13] M. Ghazvininejad, et al., "A knowledge-grounded neural conversation model," in *Proc. AAAI Conf. Artificial Intelligence*, vol. 32, no. 1, 2018.

[14] J. Yin, et al., "Neural generative question answering," *arXiv preprint arXiv:1512.01337*, 2015.

[15] Y. W. Chandra and S. Suyanto, "Indonesian chatbot of university admission using a question answering system based on sequence-to-sequence model," *Procedia Computer Science*, vol. 157, pp. 367-374, 2019.

[16] N. A. Ahmad, et al., "Review of chatbots design techniques," *Int. J. Computer Applications*, vol. 181, no. 8, pp. 7-10, 2018.

[17] R. Jonnala, et al., "Using large language models in public transit systems, San Antonio as a case study," *arXiv preprint arXiv:2407.11003*, 2024.

[18] S. Dechand, et al., "In encryption we don't trust: The effect of end-to-end encryption to the masses on user perception," in *Proc. IEEE European Symp. Security and Privacy (EuroS&P)*, IEEE, 2019.

[19] N. Tyagi and B. Bhushan, "Demystifying the role of natural language processing (NLP) in smart city applications: background, motivation, recent advances, and future research directions," *Wireless Personal Communications*, vol. 130, no. 2, pp. 857-908, 2023.