VIVEKANAND EDUCATION SOCIETY'S INSTITUTE OF TECHNOLOGY

An Autonomous Institute Affiliated to University of Mumbai

Department of Computer Engineering

Project Report on

AEROVOICE: Automatic Speech Recognition for ATC Communication

In partial fulfilment of the Fourth Year, Bachelor of Engineering (B.E.) Degree in Computer Engineering at the University of Mumbai

Academic Year 2024-2025

Submitted by

Dhruva Chaudhari - D17A 03

Preethika Shetty - D17A 58

Anurag Shirsekar - D17A 59

Sneha Tanna - D17A 62

Project Mentor

Mrs. Lifna C.S

(2024-2025)

VIVEKANAND EDUCATION SOCIETY'S INSTITUTE OF TECHNOLOGY

Department of Computer Engineering

Certificate

This is to certify that **Dhruva Chaudhari (D17A 03), Preethika Shetty (D17A 58), Anurag Shirsekar (D17A 59), Sneha Tanna (D17A 62)** of Fourth Year Computer Engineering studying under the University of Mumbai have satisfactorily completed the project on "**AEROVOICE: Automatic Speech Recognition for ATC Communication**" as a part of their coursework of PROJECT-II for Semester-VIII under the guidance of their mentor **Mrs. Lifna C.S** in the year 2024-25.

This project report entitled **AEROVOICE: Automatic Speech Recognition for ATC Communication** by ***Dhruva Chaudhari, Preethika Shetty, Anurag Shirsekar, Sneha Tanna*** is approved for the degree of **B.E. Computer Engineering.**

Programme Outcomes	Grade
PO1,PO2,PO3,PO4,PO5,PO6,PO7, PO8, PO9, PO10, PO11, PO12 PSO1, PSO2	

Date:

Project Guide:

Project Report Approval

For

B. E (Computer Engineering)

This project report entitled **AEROVOICE: Automatic Speech Recognition for ATC Communication** by ***Dhruva Chaudhari, Preethika Shetty, Anurag Shirsekar, Sneha Tanna*** is approved for the degree of **B.E. Computer Engineering.**

Internal Examiner

External Examiner

Head of the Department

Principal

Date:

Place:

Declaration

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Dhruva Chaudhari (03)

Preethika Shetty(58)

Anurag Shirsekar(59)

Sneha Tanna(62)

Date:

Acknowledgement

We are thankful to our college Vivekanand Education Society's Institute of Technology for considering our project and extending help at all stages needed during our work of collecting information regarding the project.

It gives us immense pleasure to express our deep and sincere gratitude to **Mrs. Lifna C.S** (Project Guide) for her kind help and valuable advice during the development of project synopsis and for her guidance and suggestions.

We are deeply indebted to Head of the Computer Department **Dr. (Mrs.) Nupur Giri** and our Principal **Dr. (Mrs.) J. M. Nair ,** for giving us this valuable opportunity to do this project.

We express our hearty thanks to them for their assistance without which it would have been difficult in finishing this project synopsis and project review successfully.

We convey our deep sense of gratitude to all teaching and non-teaching staff for their constant encouragement, support and selfless help throughout the project work. It is a great pleasure to acknowledge the help and suggestion, which we received from the Department of Computer Engineering.

We wish to express our profound thanks to all those who helped us in gathering information about the project. Our families too have provided moral support and encouragement several times.

Computer Engineering Department

COURSE OUTCOMES FOR B.E PROJECT

Learners will be to,

Course Outcome	Description of the Course Outcome
CO 1	Able to apply the relevant engineering concepts, knowledge and skills towards the project.
CO2	Able to identify, formulate and interpret the various relevant research papers and to determine the problem.
CO 3	Able to apply the engineering concepts towards designing solutions for the problem.
CO 4	Able to interpret the data and datasets to be utilised.
CO 5	Able to create, select and apply appropriate technologies, techniques, resources and tools for the project.
CO 6	Able to apply ethical, professional policies and principles towards societal, environmental, safety and cultural benefit.
CO 7	Able to function effectively as an individual, and as a member of a team, allocating roles with clear lines of responsibility and accountability.
CO 8	Able to write effective reports, design documents and make effective presentations.
CO 9	Able to apply engineering and management principles to the project as a team member.
CO 10	Able to apply the project domain knowledge to sharpen one's competency.
CO 11	Able to develop a professional, presentational, balanced and structured approach towards project development.
CO 12	Able to adopt skills, languages, environment and platforms for creating innovative solutions for the project.

Index

LIST OF FIGURES

LIST OF TABLES

Abstract

“This paper introduces AeroVoice: an innovative system developed for the application of ASR in Air Traffic Control communications. Conventional manual transcription and analytical techniques prove insufficient in dealing with the demands of real-time operations and have other drawbacks in light of increasing complexity and volume in aviation exchanges. Due to the use of Automatic Speech Recognition (ASR) and Natural Language Processing (NLP) in AeroVoice, ATC voice signal can now be transcribed and analysed with high accuracy and efficiency. Flight orders and callsigns, runway assignments and information about departures from normal operations can be identified and extracted by the system. The above processes are automated in AeroVoice, which enhances the awareness of the controllers and pilots, as well as reducing cognitive load and contributes to the safety operations. The advanced algorithms are invoked in AeroVoice for different accents, noise levels, or pacing making it an effective and possible solution in the future of aviation voice communications. This paper explores AeroVoice's technical design, observes its efficiency in different working conditions, and outlines enhancements to enhance the system’s accuracy and validity.”

Keywords: Air traffic Control (ATC), Automatic Speech Recognition (ASR), Speech-to-text, NLP

Chapter 1: Introduction

This chapter provides an overview of the motivation for selecting this project. It highlights the shortcomings of the current system, outlines the challenges faced by citizens, and concludes by emphasizing the project's significance.

1.1. Introduction

The aviation industry has been booming in the recent past and this has brought in extra traffic and in turn additional communication load to be met by the Air Traffic Controllers (ATCs). The smooth communication between pilots and ATCos remains crucial in flight safety, but there are barriers like background noise, accent differences, and differences in covering areas they use and acronyms. Currently used manual transcription methods contain errors leading to extra burdens on ATCos and, in some instances, compromising operational safety.

To tackle these issues we present AeroVoice, a cutting-edge ASR model trained specifically for ATC communication. AeroVoice implements speech recognition functionality that ensures correct textual or voice transformation of in-flight pilot-controller conversation in real-time from the air traffic communication channel. Apart from transcription, NLP is used to convert some of the important information from the conversations such as flight instructions, callsigns and deviations, thus eliminating any probability of misunderstanding. Automating the analysis of communications AeroVoice improves the decision making related to situations for ATCos, giving them more time to make better decisions about the situation that is taking place.

Moreover, AeroVoice plays an active role in safety management by using functions such as automatic reporting of an incident and recognition of callsigns, which highlights any irregularities for further action. This leads to a more effective and safer ATC environment where several numbers of traffic situations can be handled properly with less mental workload of the controllers. Designed specifically for aviation stakeholders, AeroVoice's ultimate vision revolves around recreating the air traffic communication industry by drastically raising its efficiency and ensuring a safer airspace.

1.2. Motivation

This project derives its purpose from the essential requirement to boost air traffic control (ATC) communication capabilities under challenging weather conditions because several recent accidents have exposed the possible hazards arising from communication breakdowns and operational difficulties.

Low visibility conditions force controllers to depend more on radio communication while maintaining positions by eye because this method creates severe safety risks through misunderstood commands. When thunderstorms or powerful winds occur the ambient noise disrupts radio communication which results in message disruption and missed communication.

Flight equipment breakdowns together with elevated workload pressures and demands from difficult flight operations increase error probabilities. Serious safety issues including near-miss incidents and runway excursions have emerged from the challenges observed at Mumbai and Chennai airports.

We will manage air traffic management risks more effectively through better communication procedures as well as strong backup systems to boost aviation safety levels in the future. The project presents itself as both timely and necessary in order to establish a more secure and efficient aviation environment.

1.3. Problem Definition

Modern air traffic controller workload intensifies because of rising aerial traffic that creates elaborate communication requirements. Flight instructions along with other critical information require manual transcription into digital systems because of which both errors and increased workload become risks. Most flight and air traffic official conversations are transmitted through Radio channels. Real-time transcription of air traffic communications demands an Automatic speech recognition(ASR) system that demonstrates accurate technical capability. The automatic transcription process of air traffic control (ATC) communications would enhance safety within the system while simultaneously boosting operational effectiveness and facilitating compliance evaluation through automated systems and improving air traffic controller training program.

1.4. Existing Systems

Air traffic communication has traditionally relied on Very High Frequency (VHF) voice transmission, although it has proved quite efficient its shortcomings are associated with misunderstanding beside frequency congestion. From a human factors perspective, this reliance on verbal communication necessitated constant vigilance from both pilots and Air Traffic Controllers (ATC), and frequently resulted in delays and/or inaccuracies – especially in messages rated as not safety critical. These challenges offered disconnection in the flow of communication over aviation topics when numbers were high or when clarity was important.

To overcome with some of these limitations, new system named Controller-Pilot Data Link

Communication (CPDLC) was initiated. Non-critical communications can be effectively dealt with by the following pre-coordinated text based messages to thus release VHF channels for more important messages. Yet, while CPDLC has benefits connected with shifting most non-emergency communications, the system also has some disadvantages. The possibility of an inaccurate message if it is sent before the real time, the message security of the system is effectively questionable in certain operational environments. Nonetheless, CPDLC also retains a high degree of human-interface involvement in message logging, tracking, and error correction, which may contain the risk of human error. However, such improvements are still insufficient to answer the need for an enhanced, real-time, and to some extent – automated communication system within aviation. To overcome these aforementioned problems, new approaches, such as AeroVoice, have been developed with the purpose to enhance the specificity of air traffic communication by means of automating transcription, analysis, and error identification of the ATC messages.

1.5. Lacuna of the Existing System

1. Air traffic control (ATC) communication operates with multiple key problems that challenge the efficiency of Controller-Pilot Data Link Communications (CPDLC). CPDLC provides dependable text messaging through a reliable system yet its installation exists only partially across all air traffic management systems. The current system depends mainly on voice communication for real-time instructions and this sustained practice results in ongoing extensive usage of traditional voice procedures. Human transcription errors and diverse accents pose problems to voice communication because they result in failed interpretations and confused understanding primarily during critical times.
2. The main issue with voice communication is its susceptibility to noise disturbances. Noise developed outside the cockpit plane and engine sounds as well as airport noise pollution can cause radio communications to be less distinct thus increasing the chance of vital information receiving misunderstanding. The current system lacks sufficient noise management technologies because CPDLC does not provide full substitution of voice-based communication for the immediate response needs in dynamic flight conditions.
3. Part of the problem with CPDLC consists of its lack of complete compatibility with existing voice communication networks. The text-based communication method provided by CPDLC offers reliable messaging but operators often refrain from using it because of inconsistent sectorwide application. Air traffic controllers experience difficulty because the two systems operate independently of each other requiring them to switch manually from voice communication to data link transmissions. High traffic situations combined with complicated airspace areas create performance issues because of decreased decision-making speed and increased workload.

4. Current voice communication methods do not include features that provide instant transcription or detect anomalies. CPDLC conducts automatic message tracking and logging yet voice communications require manual transcription that increases the likelihood of wrong information transmission. Current air traffic safety operations face a significant risk because no real-time automated system exists to detect potential communication errors.

1.6. Relevance of the Project:

The AeroVoice initiative provides essential solutions to the major communication management issues which air traffic controllers (ATCs) experience with pilots during real-time interactions. Licensed air traffic controllers currently need to handle advanced communication tasks in stressful settings that also present noisy conditions. The process of handwriting audio conversations involves excessive time usage together with a high probability of mistakes that intensifies their workload with potential miscommunication risks. The project delivers a solution through automated transcription to enhance operational efficiency and decrease ATC mental workload.

AeroVoice makes use of Automatic Speech Recognition (ASR) together with Natural Language Processing (NLP) and machine learning capabilities to execute precise real-time transcription and analysis of ATC conversations. The improved situational awareness created by this solution helps controllers making critical decisions in unsafe situations. The system accesses various accents with expert terminology while still performing in noisy settings thus enabling ATCs to work with confidence regarding quick and precise transcriptions. The innovation improves safety because it minimizes misunderstandings and provides enhanced monitoring of compliance and delivers more effective controller training programs.

Chapter 2: Literature Survey

A. Overview of literature survey:

The chapter analyzes published research about Automatic Speech Recognition (ASR) and Natural Language Understanding (NLU) in Air Traffic Control (ATC) systems by focusing on the ATCO2 project that established the largest public database for ATC speech recognition enhancement. The review covers multiple ASR algorithms starting with Wav2Vec 2.0 and hybrid solutions alongside their performance outcomes against background noise and accent fluctuations. Air India pilots have discussed the necessary technological capabilities of CPDLC which face resistance due to the fact that multiple aircraft models do not support the system.

B. Related Works

2.1. Research Papers :

1. ***Wang, Zhuang, et al. "Enhancing air traffic control communication systems with integrated automatic speech recognition: models, applications and performance evaluation." Sensors (Basel, Switzerland) 24.14 (2024).***
    a) **Abstract of the research paper:** Research investigates how ASR systems should be implemented in ATC operations for improved controller and pilot communication. The paper examines existing ASR applications together with relevant corpora and models before suggesting a deep learning evaluation method for performance enhancement. Future research recommendations outline technical solutions which address present challenges to establish ASR systems for ATC applications.
    b) **Inference drawn:** Effective ASR depends on corpora including the Air Traffic Control Complete Corpus and ATCSpeech Corpus according to this study. GMMs and DNNs offer two approaches to enhance accuracy which are examined in detail. The recognition process faces flaws that affect situational awareness because reliable recognition requires Concept Error Rates below 5% together with Command Error Rates below 3%.
2. ***Zuluaga-Gomez, Juan, et al. "Lessons learned in transcribing 5000 h of air traffic control communications for robust automatic speech understanding." Aerospace 10.10 (2023): 898.***
    a) **Abstract of the research paper:** Air traffic controllers (ATCos) must use voice

communications with pilots as their fundamental method yet ATCos often make mistakes with this essential system. The ATCO2 project meets the requirement of extensive annotated datasets by launching real-time transcription and collection of ATC audio data. This paper assesses automatic speech recognition (ASR) development by delivering public dataset results with an achieved word error rate (WER) of 17.9 percent. Beyond 10 airports the released transcribed speech reaches 5000 hours to improve comprehension of ATC communications.

b) **Inference drawn:** The ATCO2 project develops ATCO2-T Dataset through Automatic Speech Recognition technology in air traffic control systems using 5,281 hours of speech data. The ASR system reaches high accuracy by using speaker diarization and transcription methods which yield a 60% to 80% F1-Score for speaker role detection together with a 20% Jaccard Error Rate. The Named Entity Recognition system achieves recognition accuracy that reaches 97% for callsigns and 87% for commands. This system requires immense processing strength which makes it difficult to use in real-time operations within noisy air traffic control facilities.

3. ***Zuluaga-Gomez, Juan, et al. "Atco2 corpus: A large-scale dataset for research on automatic speech recognition and natural language understanding of air traffic control communications." arXiv preprint arXiv:2211.04054 (2022)***

a) **Abstract of the research paper:** The study presents ATCO2 as a new speech recognition and natural language understanding dataset focused on improving air traffic control systems. The ATCO2 corpus incorporates 4 hours of transcribed ATC speech that features expert annotations together with 5281 hours of unlabeled ATC audio recording during which automatic transcription has been generated. Spoken ATC and their associated text have been transcribed within the one-hour subset available for free. Researchers can utilize ATCO2 to advance ASR and NLU investigation for both air traffic control applications and other fields.

b) **Inference drawn:** The ATCO2 corpus contains over 5,000 hours of recorded audio with 4 hours of manual annotations that enhances ASR and NLU capabilities in ATC applications. The approach minimizes word errors while enabling NER and speaker role detection tasks which reach an F1 score evaluation of 0.97 and 0.82 respectively. Data refinement remains essential because the technique faces difficulties when operating in noisy environments together with its dependence on pseudo-labels.

4. ***García, Raquel, et al. "Automatic flight callsign identification on a controller working position: Real-time simulation and analysis of operational recordings." Aerospace 10.5 (2023): 433.***

a) **Abstract of the research paper:** An evaluation on the application of automatic speech recognition technology for air traffic management contexts explores communication enhancements between ATCos and FCs. The PJ.10-W2-96 initiative under the SESAR2020 project was developed by Enaire, Indra, Crida together with EML Speech Technology for flight callsign recognition reaching up to 84-87% accuracy for ATCos and 49-67% for FCs while demonstrating potential for enhanced performance.

b) **Inference drawn:** The research project improves ATCos' situational perception by employing ASR for call sign identification after training the system using 1,000 hours of audio data. Research analysis resulted in an 84% recognition rate for ATCos along with 67% recognition rate for FCs while achieving exceptional performance at 98% in detecting numbers from 11 to 99. The system faces problems with language variations and background sounds which demonstrate the requirement to improve its operation.

5. ***Zuluaga-Gomez, Juan, et al. "How does pre-trained wav2vec 2.0 perform on domain-shifted asr? an extensive benchmark on air traffic control communications." 2022 IEEE Spoken Language Technology Workshop (SLT). IEEE, 2023.***

a) **Abstract of the research paper:** This research evaluates self-supervised acoustic models Wav2Vec 2.0 and XLS-R for automatic speech recognition (ASR) in air traffic control domain when operated under domain shift conditions. The models operate on ATC databases with signal-to-noise ratios between 5 dB and 20 dB which leads to WER improvements between 20% and 40% compared to hybrid ASR baselines. The paper discusses WER performance in scarce resource settings and gender bias appearance within one data collection.

b) **Inference drawn:** This research evaluates Wav2Vec 2.0 functionality for domain-shifted ASR using minimal training data from private recordings of NATS (18 hours) and ISAVIA (14 hours) in combination with public datasets such as ATCO2, LDC-ATCC, UWB-ATCC, and ATCOSIM. CTC loss serves during fine-tuning operations while pre-training completes before the process. The recognition accuracy of LDC-ATCC reached 25.0% but UWB-ATCC delivered an upper rate of 54.6%. Some constraints of this system involve limitations such as demographic factors and generalizability as well as training data quality, long-term performance and environmental effects on automated speech recognition stability.

6. ***Yi, Lin, et al. "Identifying and managing risks of ai-driven operations: A case study of automatic speech recognition for improving air traffic safety." Chinese Journal of Aeronautics 36.4 (2023): 366-386.***

a) **Abstract of the research paper:** The research investigates technical dangers of air traffic control operational automation with focus on verbal exchanges in air traffic management systems. A case study analyzes objective risk indicators using questionnaire responses from users as well as interview results. The research demonstrates how the solution enhances operational safety and decreases controller burden through its risk warning abilities while operators require additional features to diminish system interruptions.

b) **Inference drawn:** The study revealed that participants characterized their responses using 10 key metrics: new practice familiarity delivered 81.4% success, warning credibility reached 94.3%, fast incident response rated at 93.8% and safety improvement achieved 92.3% while positive feedback was 92.8% but they faced excessive workload challenges that impacted 60.3% which led to a recommendation rate of 86.6% and false alarm tolerance exceeded 70%. The study faces various limitations because it does not address broader AI systems, it doesn't collect ratings from ATCO staff nor does it consider long-term effects or communication diversity along with ethical concerns regarding accountable systems that might show bias.

7. ***Shetty, Shruthi, et al. "Early Callsign Highlighting using Automatic Speech Recognition to Reduce Air Traffic Controller Workload." International Conference on Applied Human Factors and Ergonomics (AHFE2022). Vol. 60. No. 2022. 2022.***

a) **Abstract of the research paper:** The research designs an automatic speech recognition system to support air traffic controllers through the identification of spoken callsign data. Integrating surveillance data leads to performance improvement in the recognition system by reducing detection errors of callsigns from ATCo agents to 6.2% and pilot agents to 8.3% with the usage of surveillance data down to 2.8% and 4.5% respectively.

b) **Inference drawn:** Identification measurement and wrong command detection rates showed the following results: Isavia (CaRecR 96.3%, CaErrR 2.8%) and NATS (CaRecR 98.1%, CaErrR 1.8%) and also Fraport (CaRecR 95.7%, CaErrR 3.4%) and ANS CR Ops (CaRecR 98.2%, CaErrR 1.6%) and finally, ACG Lab

(CaRecR 86.9% and CaErrR 7.8%). The system deals with limitations that stem from its need for contextual information while showing a high error rate and restricted understanding of comprehensive instructions from controllers.

8. ***Fan, P., et al. "Speech recognition for air traffic control via feature learning and end-to-end training. arXiv 2021." arXiv preprint arXiv:2111.02654.***

a) **Abstract of the research paper:** The discussed paper presents an automatic speech recognition (ASR) system for air traffic control (ATC) which performs end-to-end training while employing feature learning frameworks. The model contains both a feature learning block with recurrent neural networks which reviews raw waveforms to mine significant features thus boosting the waveform-to-text conversion. The system operates with multilingual ASR functionality through its implementation of Chinese and English vocabularies. The evaluation on ATCSpeech yielded 6.9% character error rate which proved superior to existing baseline systems.

b) **Inference drawn:** The proposed model demonstrates superior performance compared to the baseline models Deep Speech 2, Jasper, Wav2letter++ while reaching Chinese CER of 7.6% and English CER of 8.9% and multilingual CER of 6.9%. Although effective the system demonstrates too much complexity and lacks sufficient performance benefits from SincNet which implies that simpler approaches would be more suitable. The testing process requires assessment in various noisy settings to achieve better system reliability.

9. ***Sestorp, Isak, and André Lehto. "CPDLC in practice: a dissection of the controller pilot data link communication security." (2019).***

a) Abstract of the research paper: This work evaluates the effectiveness of Controller-Pilot Data Link Communication (CPDLC) in securing air traffic communication at busy European airports. We assess CPDLC's practical usage and use software-defined radio technology to capture and decode its messages into readable text. Furthermore, we investigate potential security vulnerabilities and attack types stemming from intercepted CPDLC communications.

b) Inference drawn: Data interception occurs because of insufficient encryption measures. The entrance of unverified users into systems is possible through authentication weaknesses. The absence of proper monitoring tools may result in unmonitored breaches. Issues that emerge during integration phases have the potential to form vulnerabilities. System vulnerabilities result from the absence of software updates.

10. ***Kleinert, Matthias, et al. "Automated interpretation of air traffic control communication: The journey from spoken words to a deeper understanding of the meaning." 2021 IEEE/AIAA 40th Digital Avionics Systems Conference (DASC). IEEE, 2021.***

a) Abstract of the research paper: ASR technologies such as Google Assistant and Siri® dominate consumer devices even though they are insufficient for replacing pseudo-pilots in Air Traffic Control operations. Selection of ATCo-pilot messages demands more than recognition of words as operators must properly interpret their intended meanings. The 20 European partners developed a shared ontology structure that serves to enhance ATC instruction clarity. The paper advances the ontology by including speech abilities and introduces an algorithm for spoken word conversion into instructions.

b) Inference drawn: The automatic transcription of callsigns results in an extraction rate of 90.3% whereas the gold standard transcription maintains 97.2%. Automatic Callsign Extraction Rate (CaRecR) stands at 97.2% with human transcribers whereas it falls to 90.3% using machine-made transcripts. ATC requires additional system improvements due to language complexity as well as ASR limitations in the system.

2.2 Inferences drawn

Successful Air Traffic Control operations need controllers to maintain clear communication channels with pilots during their operations. ASR systems currently focus on three main goals of improving accuracy in communications while reducing workload and enhancing situational awareness. Research investigations have tested different ASR methodologies combined with datasets and model types to reach their best outcome.

Research shows that ASR functions best with high-quality corpora which includes four datasets: ATCO2, ATCSpeech, LDC-ATCC, and UWB-ATCC because they offer comprehensive annotated training materials. Experimental results show that Deep Neural Networks (DNNs) combined with Gaussian Mixture Models (GMMs) and Wav2Vec 2.0 generate better ASR accuracy which brings down Word Error Rates (WER) by 20-40% across domain-shifted domains. The advancements in Automatic Speech Recognition technology need improvement for noisy ATC surroundings that interfere with recognition outcomes.

ASR-based call sign identification technology shows an accuracy rate reaching 98% for numeric value detection which proves very useful to air traffic controllers. Speaker role detection combined with Named Entity Recognition (NER) improves ATC communication clarity through detection accuracy which achieves F1-scores of 0.97 for callsigns and 0.82 for speaker detection. OSAR-based tools still face two primary challenges because they need fast processing and struggle with real-time usage along with the requirement of better context analysis to avoid false interpretations.

Controller-Pilot Data Link Communication (CPDLC) suffers from security vulnerabilities which cause encryption system failures and authentication problems that allow unauthorized system entry. The sole implementation of ASR does not provide sufficient capability for ATC automation because ATC systems also require a common ontology structure to process complex spoken instructions past basic word recognition.

The potential value of ASR usage in ATC operations becomes apparent while it requires continued improvements for data quality along with processing speed and context-related interpretations. Upcoming research should work on decreasing mistakes in noise-affected surroundings while improving multilingual capability and strengthening security features for air traffic management communication systems.

2.3 Comparison with the existing systems

Aspect	**CPDLC**	**AeroVoice**
Communication Type	Text-based communication over data links	Real-time voice-based communication using ASR
Interaction Mode	Manual selection or typing of messages	Hands-free, voice-driven interaction
Message Handling	Pre-defined message templates and manual logging	Dynamic transcription and context-aware message extraction
Real-time capability	Limited real-time interaction due to message formulation delays	Near real-time transcription and response generation
Automation	Partially automated with human supervision	Highly automated with ASR + NLP pipeline
Accent and Noise Handling	Not applicable – assumes uniform message format	Designed to handle various accents, pacing, and background noise using robust algorithms

Table.1.Comparison with existing systems

Chapter 3: Requirement Gathering for the Proposed System

This chapter outlines the essential requirements for building the AeroVoice system. It begins by describing the motivation behind selecting specific functional and non-functional requirements to ensure accurate transcription, contextual awareness, and robust anomaly detection in air traffic communications. The section also addresses usability, performance, and regulatory compliance. The chapter further explores the software, hardware, and tools necessary for development and implementation, concluding with a detailed account of the technologies used in building the system.

3.1. Introduction to Requirement Gathering

The AeroVoice project requires functional and non-functional requirements in this chapter that emphasize vital capabilities which boost air traffic communication effectiveness. The system demands accurate voice communication transcription while requiring context analysis capabilities and anomaly detection systems to enhance safety operations along with quick response times.

A user interface needs to be simple to use and fast so air traffic controllers (ATCs) can gain access to transcriptions and alerts during critical situations. The system design addresses non-functional requirements together with performance standards alongside reliability standards and scalability standards and usability standards and security standards with the goal to provide smooth and efficient system operation.

System design faces constraints because of regulatory requirements alongside technological barriers and data privacy considerations that are addressed in this chapter. AeroVoice implementation requires a comprehensive evaluation of necessary hardware and software elements which serves as the conclusion of this section.

3.2. Functional Requirements

The essential functionalities the AeroVoice system needs to offer proper communication management for air traffic controllers (ATCs) form the basis of functional requirements. There are essential functions which the ATC Radio Conversation Analyzer needs to perform:

- Speech Recognition: The system needs to achieve high accuracy in transcribing air traffic control (ATC) communication streams. The system requires the ability to handle different speech accents which frequently occur in air traffic control working environments.

- Real-time Transcription: To benefit ATC decision-making the ASR model requires either real-time execution or processing close to real time.
- Natural Language Processing (NLP): Transcription analysis requires detection of important data which should include: Flight numbers, ATC commands, AND Emergency phrases.
- Context Awareness & Anomaly Detection: The system needs to examine dialogues for any signs of irregularities or breaches of standard operating procedures (SOPs). The detection system must identify any communication problems referred to as errors or inconsistencies between Air Traffic Controllers and pilots.
- Data Storage & Retrieval: A database system needs to handle efficient storage of transcribed data effectively. Users need access to historical conversation data which can serve needs of analysis together with auditing purposes as well as training requirements.
- Error Detection & Flagging: A combination of rule-based systems with machine learning algorithms must perform the detection of errors that occur during ATC communications. All flagged errors in transcriptions need to show clear visual indicators of their locations.
- User Interface & Accessibility: Real-time transcriptions along with error markers must appear through an interface which ATCs can understand easily. Users must be able to conduct searches and filters through the interface while having the ability to export transcribed conversations.
- Security & Compliance: The system needs to secure data storage and data transmission mechanisms to safeguard valuable ATC information. The system needs to fulfill all aviation safety rules enforced by FAA and ICAO together with GDPR and other relevant standard.

3.3. Non-Functional Requirements

The non-functional requirements establish the quality attributes alongside performance benchmarks which make AeroVoice operate effectively during air traffic control procedures.

- Performance: The transcription accuracy needs to operate with a Word Error Rate (WER) under 10% according to requirements. The current system testing shows a Word Error Rate of 0.087; however, improvement for this metric is needed.
- Scalability: The system architecture needs to scale efficiently across all environments while handling growing program traffic for expanded datasets alongside multiple current users without performance decreases. The designed system needs compatibility with cloud technology for adaptive deployment services.
- Reliability & Availability: The system should operate at 100% uptime since ATC communications hold essential mission critical nature. The system needs backup strategies that work automatically to avoid service interference when loads are at their highest point.
- Security & Data Protection: Secure AES-256 encryption rules all flight-related data and

transcription files during storage and their entire transmission duration. An RBAC system should activate as a protection measure against unauthorized users trying to access data. The solution needs to follow standards established by aviation regulators which include FAA, ICAO and GDPR and other relevant regulations.

- User-Friendliness & Usability: The system design should create an interface which controllers find easy to understand because it displays information in a simple and clear way even without special training. Real-time error highlighting and search filters along with customizable alerts should be implemented as part of the system to boost usability.
- Maintainability & Upgradability: The system needs to enable straightforward updates for all components including ASR models and error detection algorithms as well as UI enhancements. Future improvements to the codebase will be easier through modular design paired with proper documentation.
- Compliance & Regulatory Standards: This system needs to follow both aviation communication rules and industrial standard practices. The system needs complete compliance with the information security requirements of ISO 27001 and the data privacy standards of GDPR.

3.4. Hardware, Software, Technology and Tools Utilised

Software Requirements

1. Programming Languages and Frameworks
   - Backend Development:
     - Python (Flask or Django for web services)
     - Node.js (for APIs and back-end integration)
   - Frontend Development:
     - JavaScript (Next.js with Tailwind CSS for building a modern web UI)
   - Machine Learning & NLP:
     - Python (Hugging Face Transformers, PyTorch for Wav2Vec2 model implementation)
     - Python libraries for NLP tasks: SpaCy, NLTK
   - Speech Recognition:
     - Wav2Vec2 (pre-trained models from Hugging Face)
     - Alternative ASR libraries: DeepSpeech, Kaldi (for future exploration)
2. Data Integration and APIs
   - ATC Communication Sources: Live ATC feeds, and audio repositories (via APIs like LiveATC.net)

    - Flight Data API: FlightAware, ADS-B Exchange (optional for operational insights)
3. Development Tools
    - Version Control: Git (GitHub for managing code and collaboration)
    - Training & Experimentation: Kaggle Notebooks with T4 or P100 GPUs, Colab (for training models and experimenting with configurations)
4. Miscellaneous Tools
    - VS Code (for code editing)
    - Jupyter Notebook (for data exploration and model experimentation)
    - Google Colab (for light training and debugging)

Hardware Requirements

1. Computer/Laptop:
    - Minimum: Intel i5 processor (8th Gen or higher), 16GB RAM, SSD (256 GB or higher)
    - Ideal: GPU (e.g., NVIDIA GTX 1650 or higher) for local testing or debugging
    - Cloud GPUs (Kaggle, Google Colab) for heavy model training
2. Cloud Infrastructure:
    - Cloud-based GPUs: Kaggle Notebooks or Google Colab (T4, P100)
    - TPU access: Optional, if needed for large-scale model training
3. Internet Connectivity:
    - Stable high-speed internet for handling real-time ATC data streams, model training, and communication with cloud services

Technology Used:

The techniques employed include:

1. The Wav2Vec2-XLS-R-300M-en-ATC architectural model proved ideal due to its effective processing of various accents together with noisy inputs.
2. A specific fine-tuning process on the ATCOSIM Dataset became part of the model optimization to enhance its accuracy during ATC communication.
3. Military-grade audio preprocessing techniques improved acoustic quality which led to detailed transcription outputs when dealing with noisy situations.
4. The evaluation measure for assessing transcription accuracy utilizes Word Error Rate (WER).

Tools Used:

1. Google Colab/Kaggle Notebooks : Used for training the model on a GPU (T4) to

accelerate fine-tuning of the ASR model, ensuring efficient resource use.

2. PyTorch : Machine learning framework used for implementing and training the Wav2Vec2 model for ASR.
3. Hugging Face's Transformers Library : Loaded the pretrained Wav2Vec2 model, fine-tuned on the ATCOSIM dataset to optimize for ATC speech transcription.
4. Datasets Library : Facilitated the efficient loading, preprocessing, and management of the ATCOSIM dataset.
5. Python : Scripted the data preprocessing, model training, evaluation, and automation processes in the ASR project.

Chapter 4: Proposed Design

This chapter outlines the design approach for the AeroVoice system, which leverages speech recognition and natural language processing to enhance air traffic control (ATC) communication. It addresses the limitations of existing systems and the need for real-time, accurate transcription and understanding of radio conversations. The design includes integrating Automatic Speech Recognition (ASR) models like Wav2Vec2 for transcription and NLP techniques to extract critical information such as callsigns and deviations. It also focuses on anomaly detection to reduce manual workload, ultimately improving ATC efficiency. The chapter concludes by emphasizing the system's potential to optimize air traffic management.

4.1. Block Diagram of the proposed system

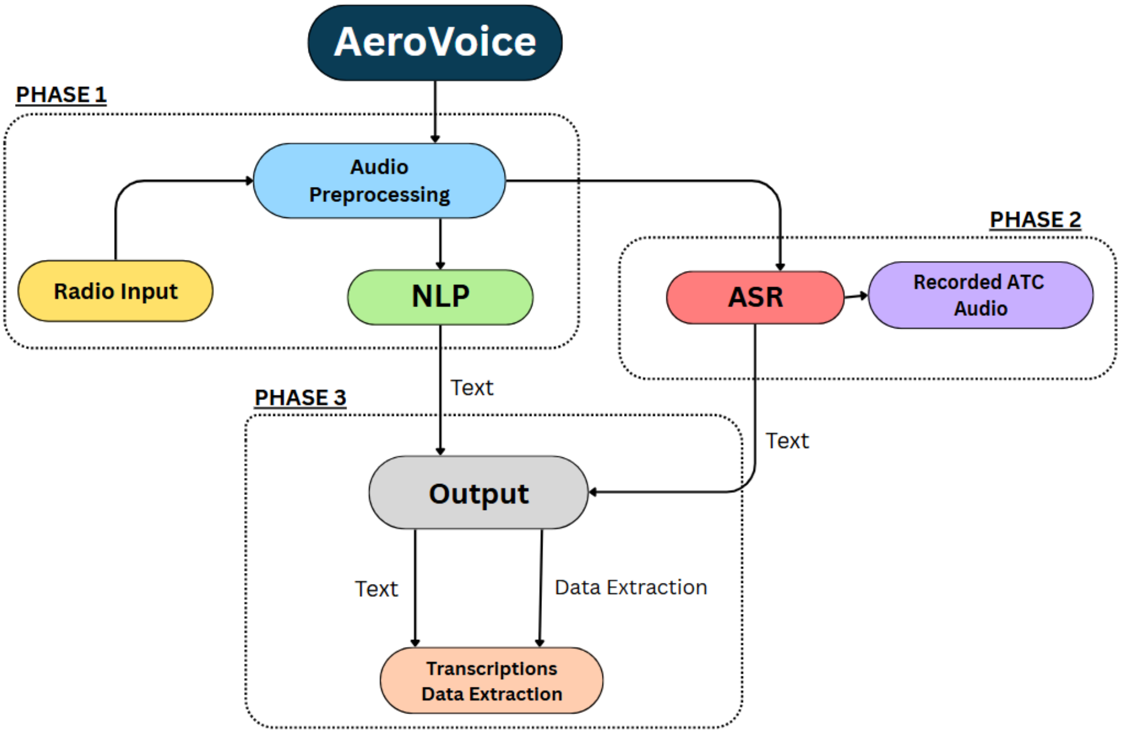


Fig.1.Block Diagram

Data Acquisition and Preprocessing:

Data Collection: This stage involves capturing live or recorded ATC audio from multiple sources like Live ATC Streams: Platforms like LiveATC.net provide real-time streams of ATC communications, Radio Receivers: Connected radios capture voice transmissions from control towers, pilots, and other aviation entities, Archived Audio: Previously recorded ATC audio is used for training and validation purposes.

Audio Preprocessing: Before analysis, the raw audio undergoes enhancement to ensure quality and model readiness:

Noise Reduction: Filters out static and background interference.

Speech Segmentation: Breaks audio into smaller, analyzable chunks.

Embedding Creation (e.g., Wave2Vec2): Converts audio signals into machine-readable representations for model input.

Speech Recognition and NLP:

ASR (Automatic Speech Recognition): Converts speech into text using AI models trained on aviation-specific datasets. The ASR engine deciphers: Pilot-controller dialogues, Call signs, Commands (e.g., altitude, heading).

NLP (Natural Language Processing): Processes the transcribed text to extract structured information: Identifies entities (e.g., aircraft IDs, altitudes, runway numbers), Flags anomalies or deviations from standard ATC phraseology, Classifies commands and responses for downstream tasks

Intelligent Analysis and Output Generation:

Pollution Detection Equivalent → ATC Interpretation & Flagging: This stage checks for the correctness and clarity of ATC communications by comparing phrases and terminology against regulatory standards: Searches the ATC terminology database, Leverages historical communication logs for context, Identifies communication errors or non-standard phrases

Prediction (Anomaly Detection & Learning):

Model Training: Historical ATC transcripts and flagged anomalies are used to improve model accuracy over time.

Model Selection: ASR and NLP performance is benchmarked using techniques like: Deep learning models (e.g., transformers, LSTMs for time-sequential data), Ensemble models for robust classification.

Report Generation and System Feedback:

Report Generation: Live Transcripts: Delivered in real-time to improve situational awareness.

Anomaly Reports: Highlight communication gaps, missed readbacks, or safety concerns.

Visual Dashboards: Provide timelines, transcripts, and flagged issues.

Feedback Loop:

User Corrections: Operators can correct transcripts, feeding the corrections back into the system.

Model Updates: Corrections are used to improve ASR/NLP performance, reducing Word Error Rate (WER) over time. Data Storage: Securely stored with encryption (e.g., AES-256) and logging for auditing and compliance.

Benefits of Using AI/ML in ATC Communication Analysis:

Improved Communication Clarity: Reduces errors in pilot-controller exchanges through real-time monitoring.

Safety and Compliance: Early detection of miscommunications helps avoid potential incidents.

Data-Driven Aviation Oversight: Supports authorities with actionable insights from communication logs.

Continuous Learning: Feedback integration ensures that the system improves with each interaction.

Scalable Monitoring: Can be applied across airports globally for consistent analysis and compliance reporting.

4.2 Modular diagram of the system

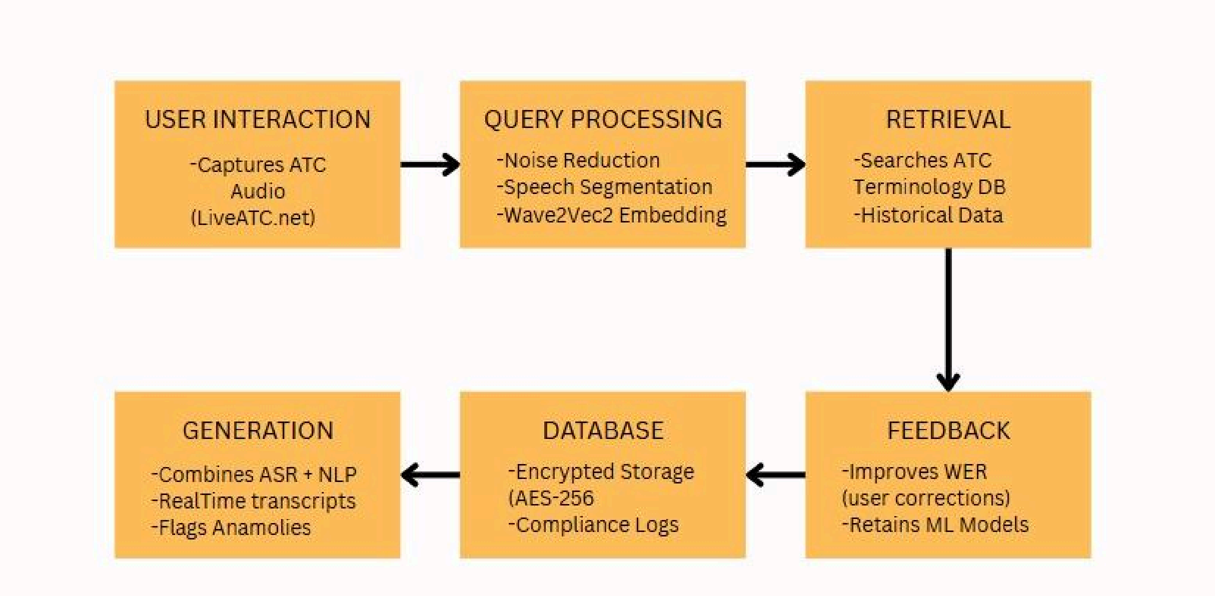


Fig.2.Modular Diagram

User Interaction: Audio Capture

Data Collection:

Live Audio Sources: The system begins by capturing real-time ATC communications from public platforms like LiveATC.net, or through direct radio feed integration. User Initiated Requests: Users or systems may request specific streams or frequencies to monitor. This step ensures that AeroVoice has a steady and dynamic flow of communication data, which is crucial for both real-time processing and long-term training.

Query Processing: Audio Preprocessing

Audio Enhancement & Preparation: Noise Reduction: Filters static, engine noise, and background chatter common in ATC environments. Speech Segmentation: Breaks down continuous audio into digestible segments based on speaker changes or pauses. Wave2Vec2 Embedding: Converts raw speech into dense, contextualized vectors that preserve acoustic and linguistic features, enabling accurate downstream speech-to-text conversion. This stage ensures the audio is clear and structured for the machine learning models to process effectively.

Retrieval: Contextual Intelligence

Data Enrichment & Reference: Terminology Lookup: The system searches a pre-built ATC terminology database, helping understand specific jargon, abbreviations, and codes. Historical Data Integration: References past communication records for context—useful for confirming flight paths, call signs, and expected commands. This helps in boosting accuracy, especially in noisy or ambiguous communication scenarios.

Feedback: Learning from User Corrections

Continuous Learning and Model Improvement: WER Reduction: Feedback from users (manual corrections, annotations) helps the system reduce Word Error Rate (WER). Model Retention: The system retains and updates the ML models based on new data and corrections, enhancing future performance. This adaptive loop allows the system to grow smarter and more reliable over time.

Database: Secure and Compliant Storage

Data Management: Encrypted Storage (AES-256): All transcripts and audio logs are stored securely, ensuring data confidentiality and protection. Compliance Logs: Every action and data point is logged to maintain traceability, supporting regulatory and legal audits. This ensures the system remains trustworthy and enterprise-grade.

Generation: Real-Time Transcripts & Intelligence

Output & Interpretation: Combined ASR + NLP Engine: Leverages both audio recognition and language understanding to generate meaningful text. Real-Time Transcription: Transcripts are created on the fly and displayed to users or fed into monitoring systems. Anomaly Detection: The system flags unusual behavior, potential miscommunications, or deviations from standard protocol. This is where the insights are produced — useful for both operational decision-making and post-event analysis.

4.3 Data Flow Diagrams

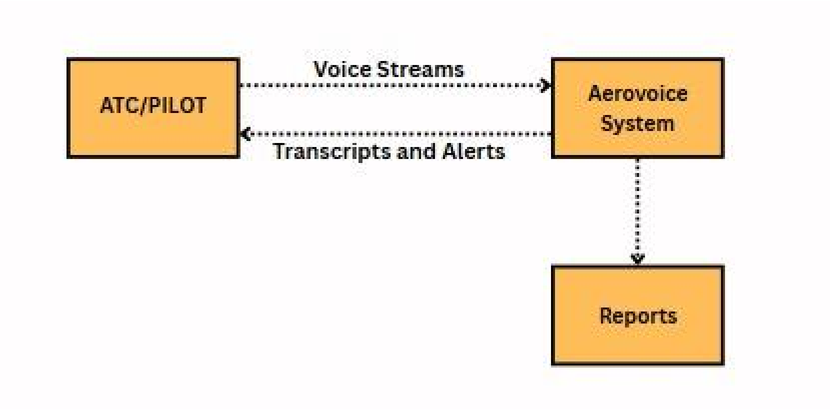


Fig.3.Level 0 DFD

Entities Involved:

- ATC/Pilot: Represents Air Traffic Controllers and Pilots who communicate using voice.
- Aerovoice System: The central processing system that handles audio data.
- Reports: The final system output in the form of readable reports.

Flow Explanation:

1. Voice Streams: Real-time voice communication from ATC or pilots is streamed into the Aerovoice System.
2. Transcripts and Alerts: The system processes these voice inputs and returns actionable items such as: Transcriptions of spoken commands, Alerts in case of detected anomalies.
3. Reports Generation: The Aerovoice System compiles data into structured Reports, which can be used for review, compliance, or training.

Purpose: This level shows the most abstract view — the Aerovoice system acts as an intermediary, converting live voice data into structured textual outputs and alerts.

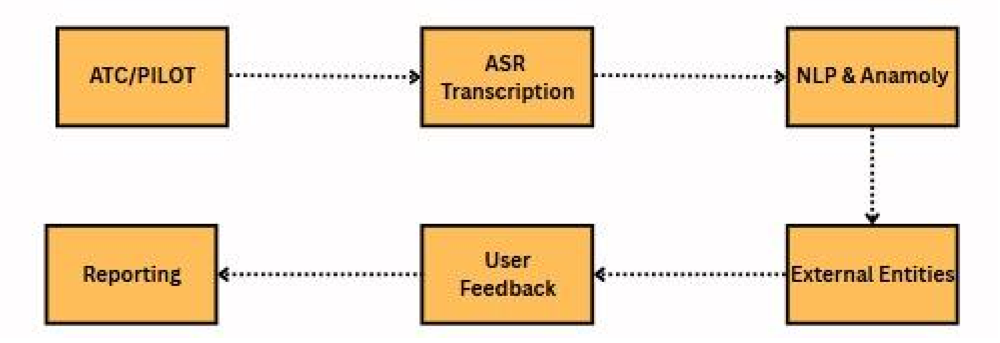


Fig.4.Level 1 DFD

Entities and Components:

- ATC/Pilot: Origin of audio communication.
- ASR Transcription: Automatic Speech Recognition module.
- NLP & Anomaly Module: Performs language processing and detects unusual patterns.
- External Entities: External authorities or systems that need reports or alerts.
- User Feedback: Mechanism for users to provide corrections or suggestions.
- Reporting: Final summarized output.

Flow Explanation:

1. Audio Input: Voice data from ATC/Pilot is sent to the ASR Transcription system.
2. Transcription Processing: The ASR converts voice into text, which is then passed to NLP & Anomaly Detection.
3. External Integration: NLP results (e.g., detected issues, commands) may be sent to External Entities like aviation authorities or alert systems.
4. User Feedback Loop: Feedback from users helps improve transcription accuracy and model performance.
5. Reporting: The output is also fed into a Reporting module for creating structured documentation or summaries.

Purpose: This level details the major functions inside the system — transcription, natural language processing, anomaly detection, external interaction, and user feedback.

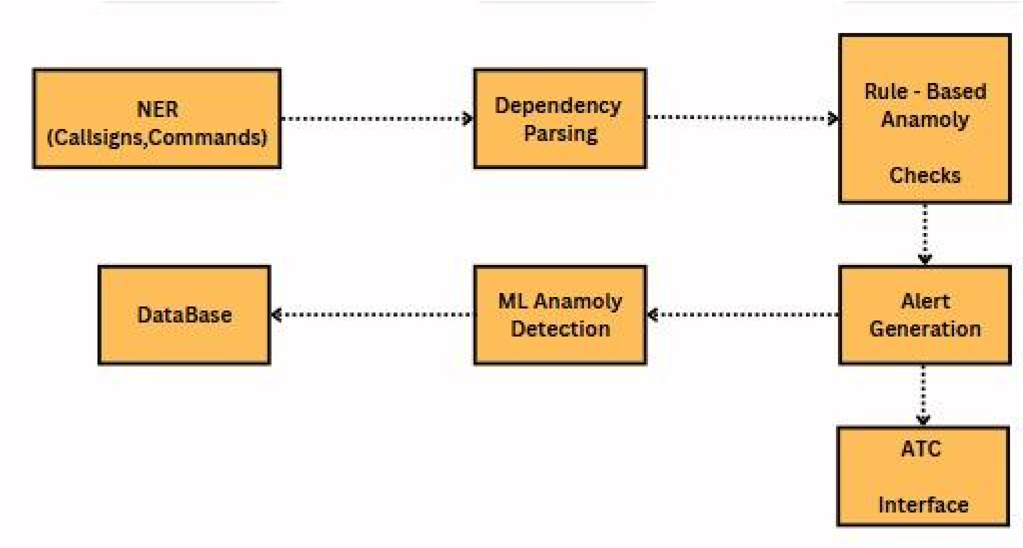


Fig.5.Level 2 DFD

Components Explained:

- NER (Named Entity Recognition): Extracts key elements from text such as callsigns, command types, and other structured entities.
- Dependency Parsing: Analyzes grammatical relationships to understand the context and intent.
- Rule-Based Anomaly Checks: Uses predefined rules to detect incorrect or suspicious communications.
- ML Anomaly Detection: A machine learning-based module that identifies complex or subtle anomalies not caught by rules.
- Alert Generation: Prepares warnings or alerts for display or communication.
- ATC Interface: Delivers the generated alerts to ATC personnel for action.
- Database: Stores all historical transcripts, alerts, and model data for training or audit purposes.

Flow Explanation:

1. NER → Dependency Parsing → Rule-Based Checks: The system extracts key terms, understands structure, and checks against known rules for issues.

2. ML Anomaly Detection: Works in parallel, learning from data and detecting non-obvious anomalies using statistical/machine learning models.

3. Alert Generation: Outputs from both anomaly detection methods feed into this module.

4. ATC Interface: Final alerts are pushed here for real-time human intervention.

5. Database Integration: All events are logged to the database for further training, compliance, and improvement.

Purpose: This level drills down into the anomaly detection workflow, showing how linguistic and statistical methods work together to keep communication safe and accurate.

4.4 Flowchart for the proposed system

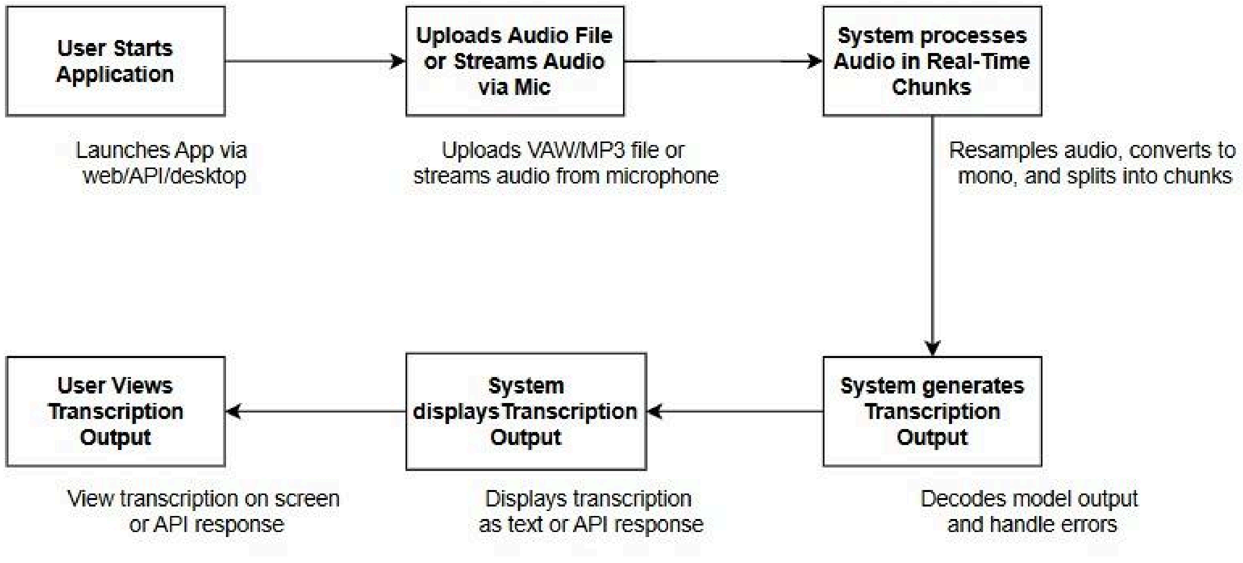

User Starts Application
Launches App via web/API/desktop
Uploads Audio File or Streams Audio via Mic
Uploads VAW/MP3 file or streams audio from microphone
System processes Audio in Real-Time Chunks
Resamples audio, converts to mono, and splits into chunks
System generates Transcription Output
Decodes model output and handle errors
System displays Transcription Output
Displays transcription as text or API response
User Views Transcription Output
View transcription on screen or API response

Fig.6.Flowchart

User Interaction & Audio Input

Step 1: User Starts Application

The user launches the transcription application through various platforms: Web Interface, API Access, or Desktop Application. This step initializes the system, preparing it to accept audio input for transcription.

Step 2: Uploads Audio File or Streams Audio via Mic

The user provides audio to the system in one of two ways: File Upload: WAV or MP3 audio files are uploaded from local storage, or Live Stream: Real-time audio is captured directly via the microphone. This step marks the beginning of the audio processing pipeline.

Real-Time Audio Processing & Transcription

Step 3: System Processes Audio in Real-Time Chunks

The system performs several preprocessing tasks: Resampling: Adjusts the audio’s sample rate for model compatibility, Mono Conversion: Converts stereo audio to a single audio channel, Chunking: Splits the audio into smaller segments for real-time analysis. This segmentation is essential for handling large files and enabling low-latency transcription.

Step 4: System Generates Transcription Output

The speech recognition model processes each audio chunk to produce text. Decoding: Converts raw model output (e.g., logits or probabilities) into readable text. Error Handling: Identifies and manages issues such as incomplete audio, silence, or noise artifacts. This step represents the core of the system where the actual transcription happens.

Display & Output Delivery

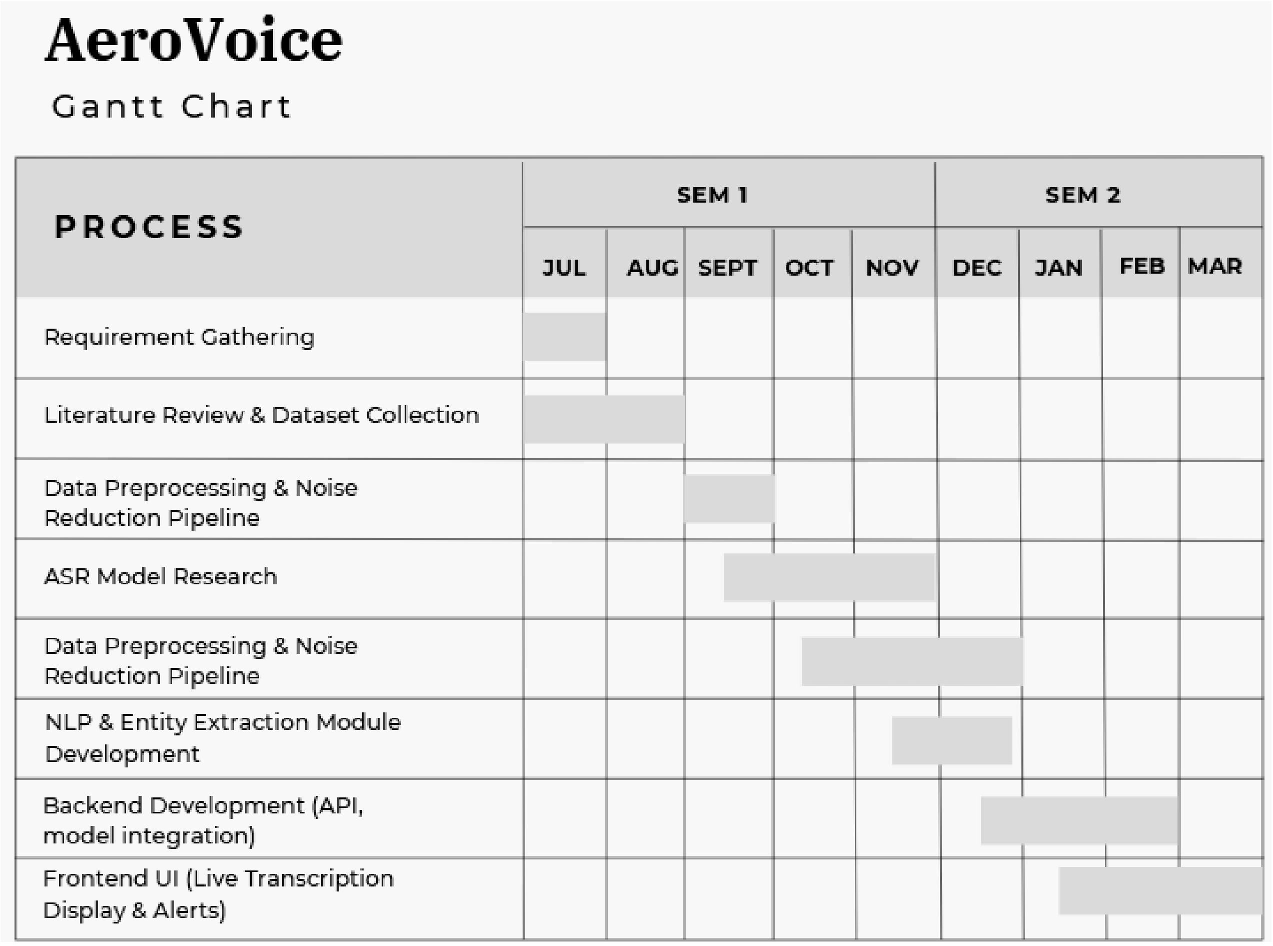
Step 5: System Displays Transcription Output

The recognized text is structured and prepared for output. Two main delivery modes: On-screen Text: For users interacting via web or desktop, and API Response: For developers using the system in integrated environments.

Step 6: User Views Transcription Output

Final step in the pipeline where the user receives the transcription results: Can read the transcribed content directly, Or parse the response in a connected application (e.g., for analytics or archiving.

4.5 Project Scheduling & Tracking using Time line / Gantt Chart



AeroVoice

Gantt Chart

PROCESS	SEM 1					SEM 2			
	JUL	AUG	SEPT	OCT	NOV	DEC	JAN	FEB	MAR
Requirement Gathering	▨								
Literature Review & Dataset Collection	▨	▨							
Data Preprocessing & Noise Reduction Pipeline			▨						
ASR Model Research			▨	▨	▨				
Data Preprocessing & Noise Reduction Pipeline				▨	▨	▨			
NLP & Entity Extraction Module Development					▨	▨			
Backend Development (API, model integration)						▨	▨	▨	
Frontend UI (Live Transcription Display & Alerts)							▨	▨	▨

Fig.7.Gantt Chart

Chapter 5: Implementation of the Proposed System

This Chapter delves into the implementation of the AeroVoice system, highlighting the development of the real-time transcription and anomaly detection functionalities. It covers the technical setup, including model selection (e.g., Wav2Vec2 for ASR), data processing, and integration with air traffic control infrastructures. The chapter also addresses challenges such as noise reduction, segmentation, and ensuring the system's robustness in various real-time scenarios. It concludes with a discussion on the system's performance evaluation and the steps taken to ensure accuracy and reliability in transcribing and understanding air traffic communication.

5.1. Methodology employed for development

AeroVoice builds their methodology through the combination of Automatic Speech Recognition (ASR), Natural Language Processing (NLP) and machine learning which leads to the development of real-time transcription together with anomaly detection functionality for Air Traffic Control (ATC) communications. The research adopts the following fundamental phases for its methodology:

1. Data Collection

The research starts with gathering authentic ATC audio communication data that consists of pilot-controller conversations. The data sources include: Publicly available aviation data repositories, the free open-source LiveATC.net serves as an online platform which offers real-time Air Traffic Controller communications, public institutions that manage airports along with aviation authorities can obtain this access through regulatory permissions, simulated ATC conversations for controlled dataset generation. The data acquisition process will gather samples which cover different accent variations alongside different speaking styles and noisy environmental factors to make the system effective in actual operational situations.

2. Data Preprocessing and Cleaning

The processing steps below will improve both the quality and application of acquired recordings.

Noise Reduction: Eliminating background noise and static interference Audio levels need normalization to create consistent input data levels.

Segmentation: Splitting long recordings into 5–10 second conversation segments The process of Text Alignment involves linking audio recordings to their corresponding written text for training purposes. This processing step puts the speech data into a condition that makes it possible to effectively train the ASR model through organized and cleaned data.

3. Automatic Speech Recognition (ASR) Model Development

A deep learning model based on Wav2Vec 2.0 architecture will perform ATC communication transcription into written text. Training will occur on the preprocessed dataset through the model development process while ensuring all points: adaptability to diverse accents and speech variations, recognition of ATC-specific terminology and aviation jargon, robustness against noisy environments common in ATC operations. The model gets enhanced using aviation jargon specifically to better detect aircraft identifiers and flight numbers alongside important instructions.

4. Natural Language Processing (NLP) and Contextual Analysis

After the speech has been transcribed into text by the ASR model, the NLP-based system will analyze the transcription to pull out and analyze important information. The following techniques will be used:

Named Entity Recognition (NER): It extracts callsigns, flight numbers, positions, and commands.

Dependency Parsing: Analyzes sentence structure to identify command relationships

Anomaly Detection: Identifies communication errors, misinterpretations, and deviations from normal ATC procedures

During this stage, not only is ATC dialogue being transcribed by the system but also interpreted into context, allowing for enhanced controllers' situational awareness.

5. System Testing and Evaluation The last stage involves large-scale testing within simulated ATC settings to assess:

ASR Model Performance: Word Error Rate of speech-to-text conversion

NLP Accuracy in flight-critical information extraction

Anomaly Detection Reliability: Efficacy of flagging in miscommunications

System Response Time and Speed: Feasibility of Real-time

It will be tested against: Speaker accent variation and speech rate variation, noisy environment with radio interference and ambient noise, complex ATC scenarios involving multiple aircraft.

Air traffic controllers' input will be utilized to further refine the system to meet real operating requirements.

5.2. Algorithms and Flowcharts for the respective modules developed

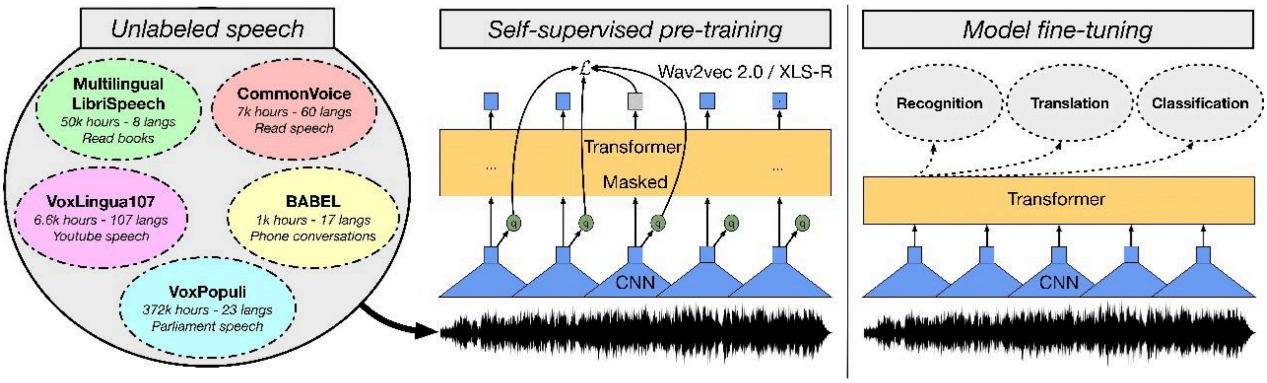


Fig.8.Wav2Vec Pipeline[18]

This shows the training and fine-tuning pipeline for Wav2Vec 2.0 (or XLS-R) — a self-supervised model for speech recognition:

Unlabeled Speech Datasets: Large corpora like LibriSpeech, CommonVoice, VoxLingua107, BABEL, and VoxPopuli are used to train models without labels.

Self-Supervised Pre-training: Audio is passed through CNN layers for feature extraction, A transformer is trained to predict masked parts of the speech representation.

Model Fine-tuning: The pretrained model is adapted for downstream tasks like: Speech Recognition, Translation, and Classification.

5.3. Datasets source and utilization:

1. Dataset Sources

The AeroVoice project utilizes the following primary datasets:

1. ATCOSIM: Air Traffic Control Simulation Speech Corpus:

- Description: A corpus comprising approximately 10 hours of non-prompted air traffic control operator speech. The recordings were captured during real-time ATC simulations in typical control room environments. citeturn0search0

- Contents:

  - Audio recordings in English from 10 non-native speakers.

  - Orthographic transcriptions with metadata on speakers and recording sessions.

- Source: Graz University of Technology (TUG) and Eurocontrol Experimental Centre (EEC).

2. UWB-ATCC: University of West Bohemia Air Traffic Control Communications Corpus:

- Description: A dataset containing approximately 20 hours of recorded communications between air traffic controllers and pilots. The speech is manually transcribed and labeled with speaker roles (pilot/controller). citeturn0search2
- Contents:
  - Audio data in 8kHz, 16-bit PCM, mono format.
  - Transcriptions with segment start and end times, speaker roles, and durations.
- Source: Department of Cybernetics, University of West Bohemia.

3. LiveATC.net Recordings:

- Description: Live audio streams from air traffic control towers worldwide, providing real-time ATC communications.
- Contents:
  - Raw audio conversations between pilots and air traffic controllers.
- Source: LiveATC.net platform.

2. Data Utilization

The datasets are employed in various components of the AeroVoice system:

- Automatic Speech Recognition (ASR) Module:
  - Utilizes the audio data to train and evaluate models that transcribe ATC communications into text with high accuracy.
- Automatic Speech Understanding (ASU) Module:
  - Processes transcriptions to extract critical information such as callsigns, altitude instructions, deviation alerts, and weather information.
  - Supports functionalities like radar label pre-filing and automatic incident detection.
- Natural Language Processing (NLP) and Machine Learning Models:
  - Train models to identify key command patterns and semantic roles in ATC conversations.
  - Classify speech segments and detect anomalies based on historical ATC data.

3. Preprocessing Steps

To ensure data quality and relevance:

- Audio Processing:
  - Apply noise reduction and silence removal techniques.
  - Perform speech segmentation and speaker diarization.
- Text Processing:
  - Conduct tokenization and Named Entity Recognition (NER) for ASU tasks.
  - Implement data augmentation strategies to enhance training data diversity for ASR models.

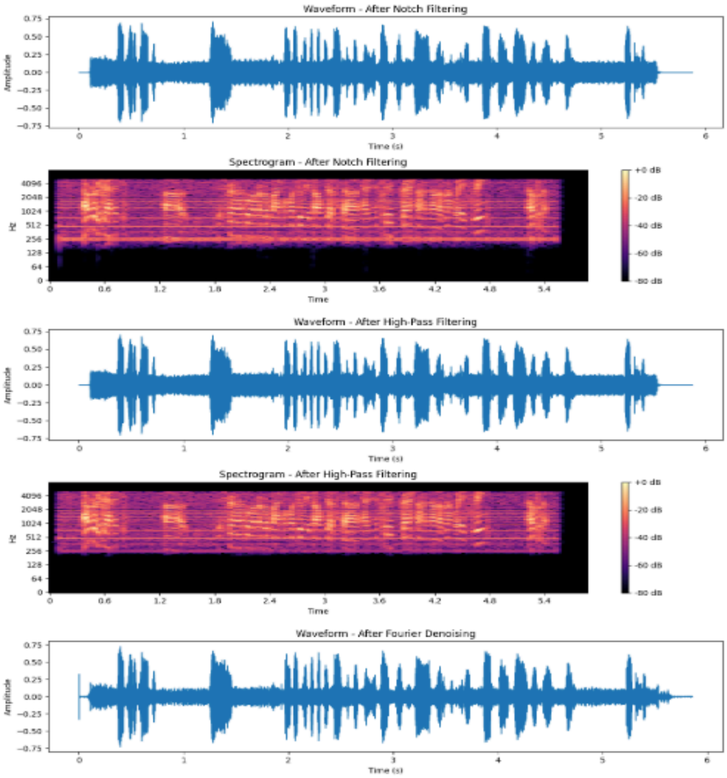


Figure 9: Screenshot of Audio Preprocessing using techniques - Notch Filtering , High Pass Filtering and Fpurier Denoising

4. Ethical Considerations

All data utilized is either publicly available or anonymized to comply with aviation data-sharing standards, ensuring privacy and confidentiality are maintained.

Chapter 6: Testing of the Proposed System

This chapter focuses on the systematic testing of the AeroVoice system. It includes both the alpha and beta testing phases. Alpha testing involved internal validation by the development team, testing key features such as Automatic Speech Recognition (ASR), Automatic Speech Understanding (ASU), and error detection. Beta testing involved external testers, including aviation professionals and ATC trainees, to evaluate the system under real-world conditions. The chapter also discusses various test cases, such as noise and overlap simulations, stress testing, and entity extraction validation to ensure the system's robustness and accuracy.

6.1. Introduction to Testing

Software testing is a critical phase in the development of any system and involves the systematic execution of test cases to evaluate the correctness, completeness, and quality of the system. For AeroVoice, a research-based project focused on enhancing speech recognition and natural language processing (NLP) for ATC communications, the testing phase aims to validate functionality, performance, and reliability across various scenarios.

The testing process provides insights into potential risks or deviations from the expected system behavior and helps stakeholders make informed decisions before deployment. By aligning test activities with project specifications, the system's alignment with functional and non-functional requirements is ensured.

6.2. Types of tests Considered

A. Pre testing phase

Alpha testing for AeroVoice involved initial internal validation of system components by the development team in a controlled setting. The focus was on evaluating core modules such as **Automatic Speech Recognition (ASR)**, **Automatic Speech Understanding (ASU)**, and **Error Detection**.

Aspects tested during alpha phase:

- **Functionality**
  - ○ Verification of key features: ASR transcription, ASU entity extraction (e.g., call signs, instructions), and error detection modules.
- **Performance**
  - ○ Testing transcription latency and ASU extraction accuracy under simulated air

traffic audio.

- **Reliability**
  - Checking the system's stability by running it continuously with varied audio inputs.
- **User Interface**
  - Reviewing the AeroVoice dashboard for transcription clarity, labeling, and timeline synchronization.
- **Compatibility**
  - Ensuring the tool operates across various devices and browsers used by ATCos and researchers.
- **Data Accuracy**
  - Comparing transcribed outputs and recognized entities against manually labeled ground truth data.
- **Fault Tolerance**
  - Evaluating the system's response to corrupted audio, speech overlaps, or communication dropouts.
- **Feedback Collection**
  - Internal testers reported usability issues, system lag, and recognition failures for further refinement.

B. Beta-Testing Phase

Beta testing involved releasing AeroVoice to a selected group of external testers including domain experts, ATC trainees, and academic collaborators. The goal was to test the system under realistic communication scenarios and receive holistic feedback.

Key components of beta testing:

- Selection of Beta Testers

  Aviation professionals, NLP researchers, and ATC training institutes.
- Deployment and Usage

  Testers uploaded live or recorded ATC audio sessions and used AeroVoice's transcription and analysis features.
- Testing Period

  The beta phase spanned 2–4 weeks to capture diverse audio inputs and communication patterns.
- Feedback Collection

  User feedback was collected via forms, focus groups, and issue trackers covering usability, clarity, and overall experience.
- Bug Reporting

Identified bugs such as misrecognition of similar call signs, incorrect punctuation in transcripts, or UI lags were logged and fixed.

6.3. Various test case scenarios considered

The following test case scenarios were designed to evaluate AeroVoice's robustness and accuracy:

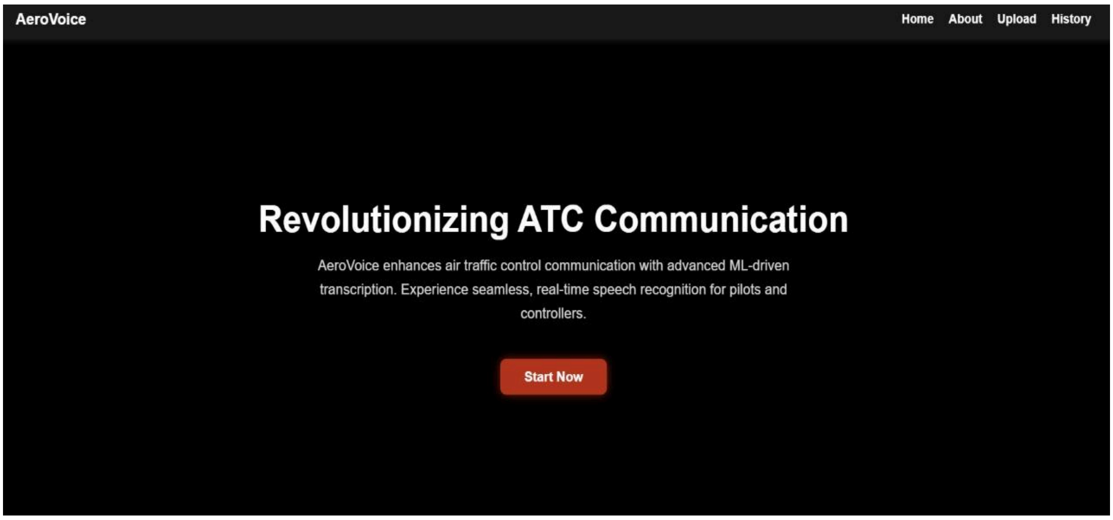
1. **Baseline Testing**
    - ○ Feed clear ATC audio clips and compare ASR and ASU outputs with verified ground truth.
    - ○ Validate identification of basic entities like callsigns, aircraft type, runway, and altitude.
2. **Noise and Overlap Simulation**
    - ○ Introduce overlapping speech, static noise, and abrupt interruptions.
    - ○ Evaluate how well the system distinguishes speakers and maintains transcription accuracy.
3. **Variable Accents and Speech Rates**
    - ○ Use ATC speech samples from different regions to test adaptability.
    - ○ Test ASR model's ability to handle fast, clipped speech or heavy accents.
4. **Stress Testing Under Continuous Load**
    - ○ Run continuous live audio feeds to simulate real-time control tower environments.
5. **Data Consistency and Timestamp Verification**
    - ○ Ensure temporal alignment between audio and transcript timeline.
    - ○ Verify accurate tagging of critical phrases like "cleared for takeoff" or "go-around."
6. **Speaker Identification Accuracy**
    - ○ Test the model's ability to differentiate between pilot and controller communications.
7. **Entity Extraction Validation (ASU)**
    - ○ Assess NLP model's capability to extract and classify key information such as wind speed, altitude changes, and runway assignments.
8. **Data Export and Storage**
    - ○ Check proper export functionality for transcript reports and JSON logs.
9. **Error Reporting and Feedback Integration**

○ Verify correct logging of transcription errors and flagging of communication anomalies.

Chapter 7: Results and Discussions

This chapter provides an in-depth look at the performance and results of the AeroVoice system. It includes detailed screenshots of backend and user interface (UI) showing the main features of the system, such as the homepage, "About" page, and upload page. The chapter also includes a discussion on performance evaluation measures, particularly focusing on the Word Error Rate (WER) to assess transcription accuracy. Additionally, it compares the results of the system with existing ASR models, highlighting the strengths and limitations of AeroVoice in handling ATC-specific speech challenges.

7.1 Screenshot of Use Interface(UI) for the system:



Revolutionizing ATC Communication

AeroVoice enhances air traffic control communication with advanced ML-driven transcription. Experience seamless, real-time speech recognition for pilots and controllers.

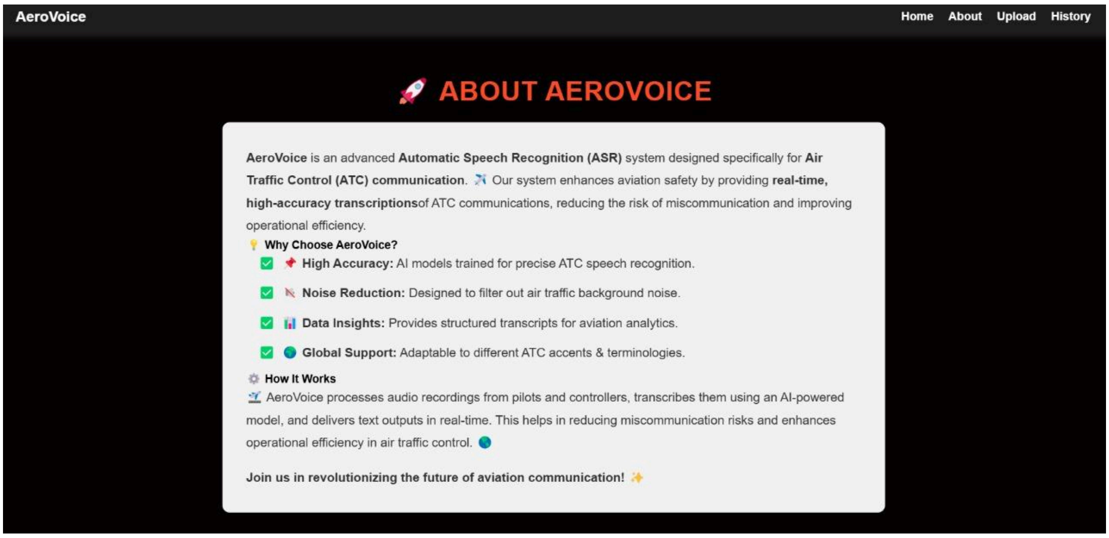
Fig.10. UI (1)

1. Homepage – "Revolutionizing ATC Communication"

This is the landing page of AeroVoice, designed to grab attention with a bold heading and a call-to-action button ("Start Now").

Purpose:

- Highlights AeroVoice’s main goal: revolutionizing air traffic communication using real-time ML-driven transcription.
- Emphasizes speech recognition for pilots and controllers, hinting at automation and operational efficiency.
- User-friendly UI to prompt interaction.



🚀 ABOUT AEROVOICE

AeroVoice is an advanced **Automatic Speech Recognition (ASR)** system designed specifically for **Air Traffic Control (ATC) communication**. ✈️ Our system enhances aviation safety by providing **real-time, high-accuracy transcriptions**of ATC communications, reducing the risk of miscommunication and improving operational efficiency.

💡 **Why Choose AeroVoice?**

- ✅ 📌 **High Accuracy:** AI models trained for precise ATC speech recognition.
- ✅ 🔇 **Noise Reduction:** Designed to filter out air traffic background noise.
- ✅ 📊 **Data Insights:** Provides structured transcripts for aviation analytics.
- ✅ 🌍 **Global Support:** Adaptable to different ATC accents & terminologies.

⚙️ **How It Works**

🛬 AeroVoice processes audio recordings from pilots and controllers, transcribes them using an AI-powered model, and delivers text outputs in real-time. This helps in reducing miscommunication risks and enhances operational efficiency in air traffic control. 🌍

Join us in revolutionizing the future of aviation communication! ✨

Fig.11.UI (2)

2. About Page – “About AeroVoice”

This section introduces the features and technical design of AeroVoice. It clearly outlines: What AeroVoice is (ASR for ATC communication), Its key advantages, How it works.

Key Features Highlighted:

- High Accuracy: AI models optimized for aviation speech.
- Noise Reduction: Filters out typical ATC background noise.
- Data Insights: Helps in analytics by structuring transcriptions.
- Global Support: Adaptable to various regional accents and terminologies.

Purpose:

- Educates users (or stakeholders) about the need for AeroVoice and how it solves current issues.
- Explains the workflow: records → transcribes → delivers text in real-time.

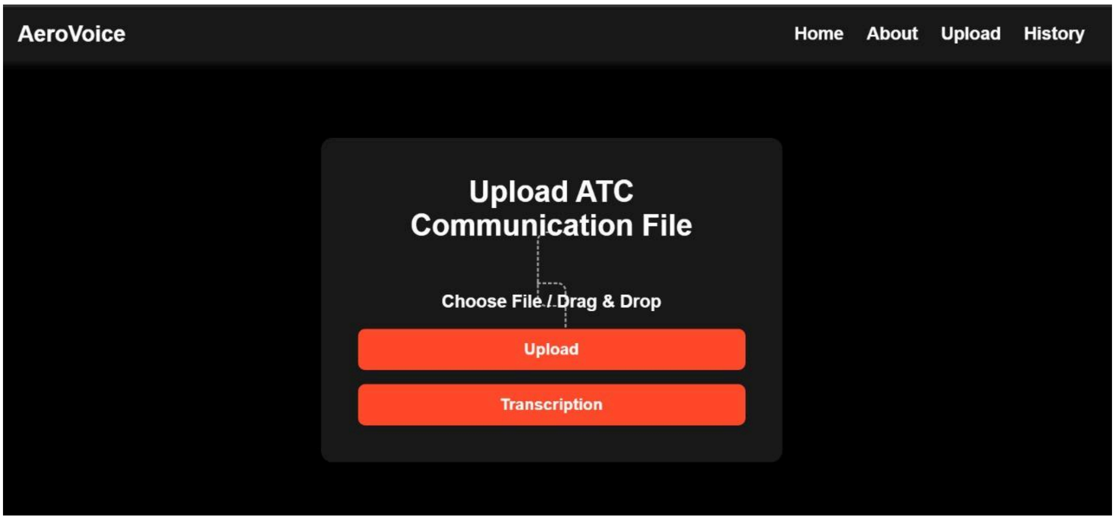
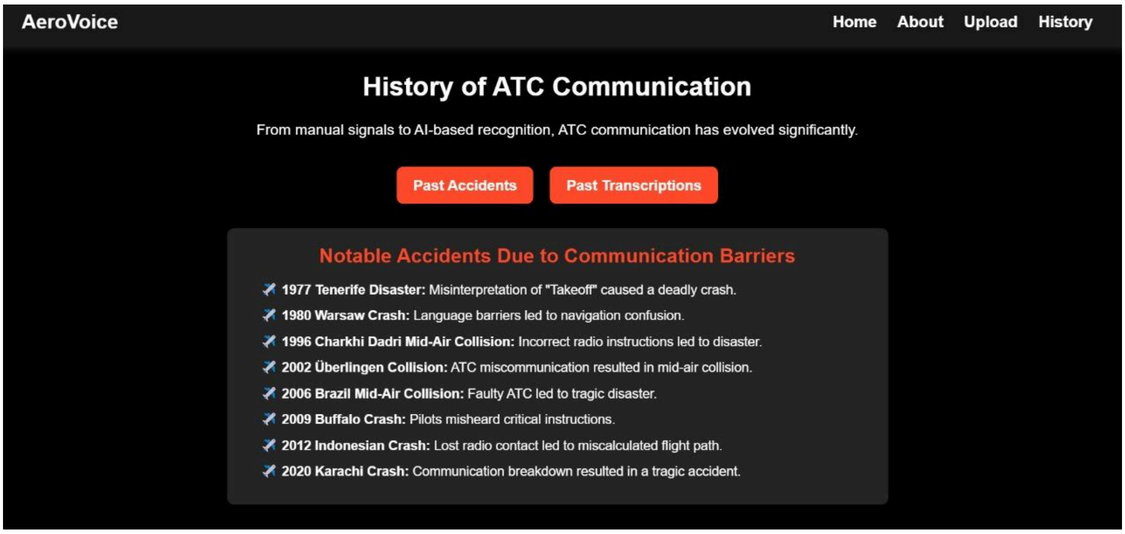


Fig.12. UI (3)



History of ATC Communication

From manual signals to AI-based recognition, ATC communication has evolved significantly.

Notable Accidents Due to Communication Barriers

✈ **1977 Tenerife Disaster:** Misinterpretation of "Takeoff" caused a deadly crash.
✈ **1980 Warsaw Crash:** Language barriers led to navigation confusion.
✈ **1996 Charkhi Dadri Mid-Air Collision:** Incorrect radio instructions led to disaster.
✈ **2002 Überlingen Collision:** ATC miscommunication resulted in mid-air collision.
✈ **2006 Brazil Mid-Air Collision:** Faulty ATC led to tragic disaster.
✈ **2009 Buffalo Crash:** Pilots misheard critical instructions.
✈ **2012 Indonesian Crash:** Lost radio contact led to miscalculated flight path.
✈ **2020 Karachi Crash:** Communication breakdown resulted in a tragic accident.

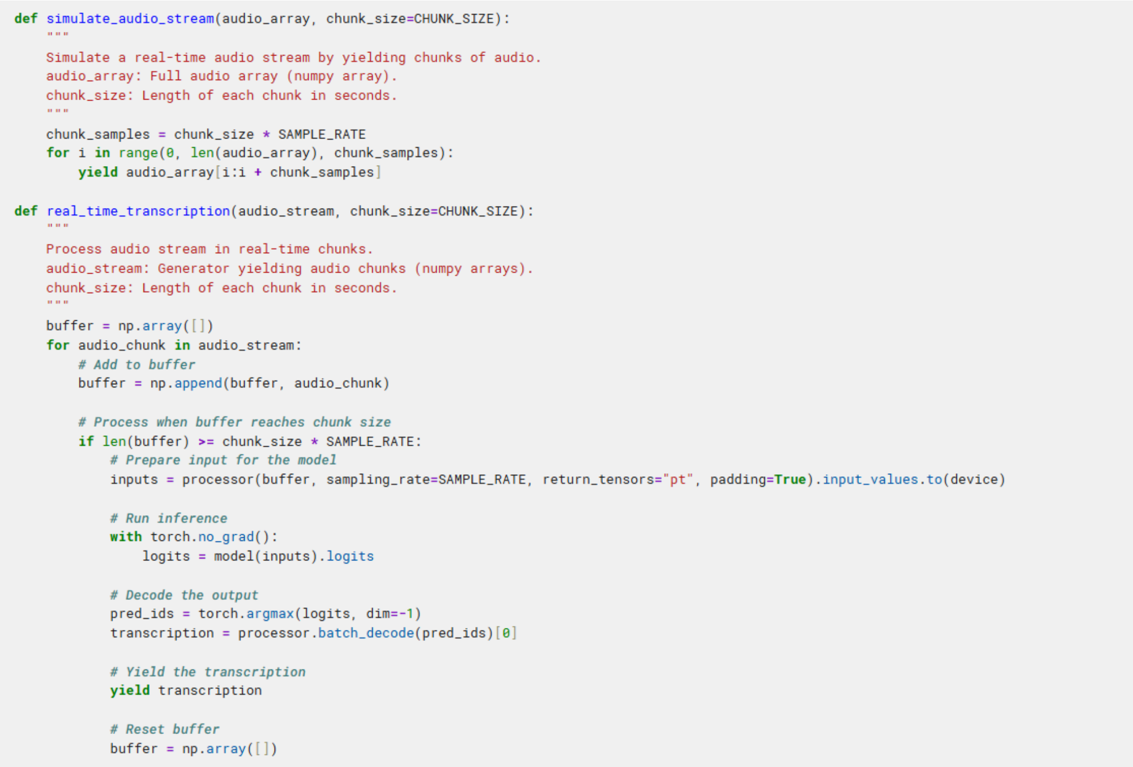
Fig.13. UI (4)

3 & 4. Upload Page – "Upload ATC Communication File"

What it shows: This is the core functional part of the AeroVoice interface, where users can: Upload an ATC audio file (via drag and drop or file picker), Click Transcription to begin the ASR process.

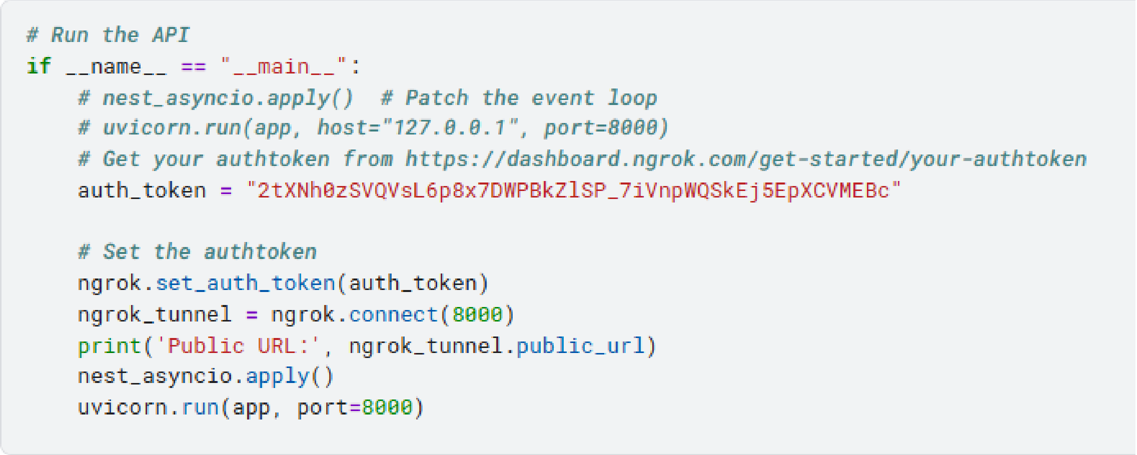
Purpose:

- Makes the system interactive and usable.
- Emphasizes simplicity for users to test AeroVoice's capabilities.
- Ensures the whole process is automated, streamlined, and accessible for both research and real-world usage.



```
def simulate_audio_stream(audio_array, chunk_size=CHUNK_SIZE):
    """
    Simulate a real-time audio stream by yielding chunks of audio.
    audio_array: Full audio array (numpy array).
    chunk_size: Length of each chunk in seconds.
    """
    chunk_samples = chunk_size * SAMPLE_RATE
    for i in range(0, len(audio_array), chunk_samples):
        yield audio_array[i:i + chunk_samples]

def real_time_transcription(audio_stream, chunk_size=CHUNK_SIZE):
    """
    Process audio stream in real-time chunks.
    audio_stream: Generator yielding audio chunks (numpy arrays).
    chunk_size: Length of each chunk in seconds.
    """
    buffer = np.array([])
    for audio_chunk in audio_stream:
        # Add to buffer
        buffer = np.append(buffer, audio_chunk)

        # Process when buffer reaches chunk size
        if len(buffer) >= chunk_size * SAMPLE_RATE:
            # Prepare input for the model
            inputs = processor(buffer, sampling_rate=SAMPLE_RATE, return_tensors="pt", padding=True).input_values.to(device)

            # Run inference
            with torch.no_grad():
                logits = model(inputs).logits

            # Decode the output
            pred_ids = torch.argmax(logits, dim=-1)
            transcription = processor.batch_decode(pred_ids)[0]

            # Yield the transcription
            yield transcription

            # Reset buffer
            buffer = np.array([])
```

Figure 14:Screenshot of code snippet for Automatic Speech Recognition (Backend)



```python
# Run the API
if __name__ == "__main__":
    # nest_asyncio.apply()  # Patch the event loop
    # uvicorn.run(app, host="127.0.0.1", port=8000)
    # Get your authtoken from https://dashboard.ngrok.com/get-started/your-authtoken
    auth_token = "2tXNh0zSVQVsL6p8x7DWPBkZlSP_7iVnpWQSkEj5EpXCVMEBc"

    # Set the authtoken
    ngrok.set_auth_token(auth_token)
    ngrok_tunnel = ngrok.connect(8000)
    print('Public URL:', ngrok_tunnel.public_url)
    nest_asyncio.apply()
    uvicorn.run(app, port=8000)
```

Figure 15: Screenshot of code snippet for FastAPI(Backend)

7.2 Performance Evaluation Measures:

The performance measures considered in the project are :

- Word Error Rate (WER): The primary metric used to assess the model's performance is the Word Error Rate (WER). WER quantifies the accuracy of the transcriptions by comparing the predicted text to the actual text. It is calculated using the formula:

  WER=(S+D+I)/N

  where:

  - S = Number of substitutions (incorrect words predicted)
  - D = Number of deletions (words missed)
  - I = Number of insertions (extra words added)
  - N = Total number of words in the reference transcript
  - A lower WER indicates better performance. In this project, the model achieved a WER of 0.085, suggesting high transcription accuracy.
- Evaluation Mode: During the evaluation phase, the model is set to evaluation mode (model.eval()), which ensures that certain layers (like dropout) behave appropriately for inference, providing a more accurate measure of performance.
- Batch Predictions: The evaluation code processes audio in batches, generating predictions without tracking gradients. This approach optimizes the evaluation speed and resource usage.
- Transcription Comparison: The predicted transcriptions are compared against the actual labels (transcripts) using the WER metric to evaluate how well the model performs in recognizing and transcribing air traffic control communications.

7.3 Input Parameters/Features considered

- Audio Data: The main input is the audio recordings from the atcosim_corpus dataset, which captures air traffic control communications.
- Transcriptions: Each audio file is paired with its corresponding text transcript, guiding the model to learn how to convert speech into written words.
- Batch Processing: Data is processed in batches, allowing the model to train efficiently on multiple audio files at the same time.
- Model Settings: Key configurations, like the learning rate and optimization method, are established before training to enhance performance.
- Duration: The duration of each audio segment, computed as segment_end_time - segment_start_time, provides insights into the length of the recordings, which can be relevant for model performance and evaluation.

7.4 Comparison of Results with Existing System

Model	Metrics	Limitations
GMM-HMM [1]	ConER: 50%, CmdER: 100%	Misses ASR errors, needs external tools.
DNN-HMM Hybrid [2]	F1: 60–80%, Callsigns: 97%, Cmds: 87.1%	High resource use, less real-time friendly.
End-to-End ASR [5]	Role F1: 0.83–0.86, WER: 22–11%	Needs big data, noise-sensitive
BiLSTM + Hybrid ASR [3]	Acc: Ctrl 84%, Crew 67%, Callsign: 58.5%	Weak on callsigns, noise, language variety.
Wav2Vec 2.0 [4]	WER: 7.7–35.8% (across datasets)	Poor generalization, needs robustness.
Deep Speech + RNN [17]	WER: 17%, Callsign F1: 0.95	Struggles with ATC jargon, accents.
DNN + MIP [6]	Safety: 92.3%, Credibility: 94.3%	Ignores bias, lacks long-term validation.
SincNet + CNN-RNN [7]	CER: 6.9–8.9%	Accurate but too complex, weak noise handling.
TDNNF + BPE [15]	WER: 7.75%, +35% with BPE	Top performer, needs accent/generalization work.

Table.2.Comparison of Results With existing System

Inference Drawn:

- Traditional vs. Deep Learning ASR

  Because the GMM-HMM models work correctly with existing ATC systems they maintain a CmdER of 100%. The callsign recognition accuracy achieved 97% success while command detection scores 87.1% and the models demand big training sets and processing strength.

- End-to-End ASR Needs Large Data but Shows High Accuracy

  The implementation of WER reconstructed at 22.3% but was actually reduced to 11.1% along with Speaker Role Detection reaching an F1 score of 0.86-0.83. The system operates in noisy ATC conditions while using pseudo-labels that introduce possible errors to the analysis.

- Hybrid ASR Models Improve Recognition but Have Limitations

  BILSTM + Hybrid ASR demonstrates controller recognition at 84% but shows difficulty when identifying flight crew messages (67%) as well as callsign strings (58.5%). Phonetic variations together with linguistic diversity within ATC speech cause issues during transmissions.

- Wav2Vec 2.0 Shows Potential but Lacks ATC-Specific Robustness

  The ATC datasets show elevated WER statistics of 58.7% in ATCO2 and 54.6% in UWB-ATCC. The system encounters difficulties due to ATC terminology, various speech accents and noisy operational conditions.

- Deep Speech + RNN & TDNNF Improve WER but Have Deployment Issues

  Deep Speech + RNN reaches a WER equal to 17% yet N-gram modeling enhances its accuracy by 26%. The WER performance of TDNNF decreased by 35% yet the system demonstrates limitations when processing accents and speakers in natural operating environments.

- CNN-Based & Risk-Detection Models Improve Safety but Are Complex

  The CER performance of SincNet + CNN-RNN amounts to 7.6% Chinese and 8.9% English making it suitable for multilingual ATC operations but challenging to implement. The combination of DNN + MIP increases ATC safety levels to 92.3% however it does not provide assessment of sustained impact or receive feedback from ATCOs.

Chapter 8: Conclusion

In this chapter, the limitations and future directions for the AeroVoice system are discussed. While the system shows significant improvements in transcription and understanding of ATC communication, it still faces challenges such as noisy audio handling, latency, and accent variability. The chapter concludes with a summary of the project's contributions and outlines future enhancements, including real-time deployment, better noise reduction, and optimized hybrid models to further improve the system's effectiveness in ATC operations.

8.1 Limitation

While the AeroVoice system demonstrates notable improvements in transcription and understanding of ATC communication, it still has a few limitations:

1. **Handling of Noisy Audio:** Although the system is trained on noisy datasets like LiveATC, it still struggles with extremely poor-quality audio that includes overlapping speech, static interference, or abrupt cut-offs. Background cockpit or tower noise can occasionally impact recognition accuracy, especially for low-volume or clipped transmissions.
2. **Latency in Response Generation:** The system introduces a slight delay in processing due to sequential stages like ASR, NLP parsing, and intent extraction. While acceptable for offline analysis and post-event evaluation, this latency may not yet meet the stringent real-time requirements of operational ATC environments.
3. **Accent and Phraseology Variability:** The model is optimized for standard aviation English and may have reduced performance when encountering strong accents, regional dialects, or informal/non-standard phraseology.
4. **Dataset Coverage Limitations:** The current training datasets do not comprehensively cover emergency scenarios, non-routine commands, or low-frequency events, which limits the model's ability to handle rare or unexpected interactions.
5. **Contextual Understanding:** The system primarily processes each transmission independently and lacks full conversational context tracking across multiple turns or overlapping dialogues between pilot and controller.

8.2 Conclusion

The AeroVoice project successfully showcases how the combination of ASR and NLP technologies enhances ATC communication efficiency operations. Through the investigation of many ASR models we established the main advantages and drawbacks of traditional systems and deep learning-based approaches when applied in ATC environments. Real-time deployment of deep learning models faces obstacles because of their limitation with dealing with noise interference and computational overhead and the variations in accents in the system.

Future efforts will focus on reducing latency through optimized streaming architectures, improving noise handling with advanced filtering techniques, and developing hybrid models tailored for ATC-specific terminology and variations. The system will benefit from powerful logging methods coupled with artificial intelligence algorithms for improving transcription accuracy and generating valuable information for air traffic control officers.

The solutions implemented at AeroVoice close the distance between speech recognition improvements and operational ATC implementation. Continuous optimization of this system and real-time deployment capabilities will produce better air traffic management results by cutting manual workloads while boosting situational awareness and improving communication accuracy throughout aviation operations.

8.3 Future Scope:

AeroVoice will get future updates through real-time ATC communication by eliminating delays and delivering quick transcriptions that require little latency. The system will use advanced noise reduction capabilities to eliminate background sounds and enhance difficult ATC voice signals. The system will receive sped-up optimization that shortens speech recognition processing duration to produce almost instantaneous transcription results. The system will establish a comprehensive logging solution which tracks flights through real-time monitoring while automatically finding errors in communication together with saving potential analysis material. AeroVoice transforms into a faster more accurate system along with improving reliability when used for ATC operations.

References

[1] Wang Z, Jiang P, Wang Z, Han B, Liang H, Ai Y, Pan W. Enhancing Air Traffic Control Communication Systems with Integrated Automatic Speech Recognition: Models, Applications and Performance Evaluation. *Sensors*. 2024; 24(14):4715. https://doi.org/10.3390/s24144715

[2] Zuluaga-Gomez J, Nigmatulina I, Prasad A, Motlicek P, Khalil D, Madikeri S, Tart A, Szoke I, Lenders V, Rigault M, et al. Lessons Learned in Transcribing 5000 h of Air Traffic Control Communications for Robust Automatic Speech Understanding. *Aerospace*. 2023; 10(10):898. https://doi.org/10.3390/aerospace10100898

[3] García R, Albarrán J, Fabio A, Celorrio F, Pinto de Oliveira C, Bárcena C. Automatic Flight Callsign Identification on a Controller Working Position: Real-Time Simulation and Analysis of Operational Recordings. *Aerospace*. 2023; 10(5):433. https://doi.org/10.3390/aerospace10050433

[4] Prasad, A., Nigmatulina, I., Sarfjoo, S., Motlicek, P., Kleinert, M., Helmke, H., Ohneiser, O., & Zhan, Q. (2022). How Does Pre-trained Wav2Vec 2.0 Perform on Domain Shifted ASR? An Extensive Benchmark on Air Traffic Control Communications. *ArXiv*. https://arxiv.org/abs/2203.16822

[5] J. Zuluaga-Gomez *et al.*, "ATCO2 corpus: A Large-Scale Dataset for Research on Automatic Speech Recognition and Natural Language Understanding of Air Traffic Control Communications," *arXiv.org*, 2022. https://arxiv.org/abs/2211.04054 (accessed Apr. 01, 2025).

[6] Lin, Yi & Ruan, Min & Cai, Kunjie & Li, Dan & Zeng, Ziqiang & Li, Fan & Yang, Bo. (2022)."Identifying and managing risks of AI-driven operations: A case study of automatic speech recognition for improving air traffic safety," Chinese Journal of Aeronautics, Aug. 2022, doi: https://doi.org/10.1016/j.cja.2022.08.020.

[7] P. Fan, D. Guo, Y. Lin, B. Yang, and J. Zhang, "Speech recognition for air traffic control via feature learning and end-to-end training," arXiv.org, 2021. https://arxiv.org/abs/2111.02654

[8] S. Shetty, H. Helmke, M. Kleinert, and O. Ohneiser, "Early Callsign Highlighting using Automatic Speech Recognition to Reduce Air Traffic Controller Workload," *AHFE international*, Jan. 2022, doi: https://doi.org/10.54941/ahfe1002493.

[9] I. Sestorp and A. Lehto, "CPDLC in Practice : A Dissection of the Controller Pilot Data Link Communication Security," Dissertation, 2019.

[10] M. Kleinert et al., "Automated Interpretation of Air Traffic Control Communication: The Journey from Spoken Words to a Deeper Understanding of the Meaning," 2021 IEEE/AIAA 40th Digital Avionics Systems Conference (DASC), San Antonio, TX, USA, 2021, pp. 1-9,

doi: 10.1109/DASC52595.2021.9594387

[11] H. Glaser-Opitz and L. Glaser-Opitz, "Evaluation of CPDLC and voice communication during approach phase," 2015 IEEE/AIAA 34th Digital Avionics Systems Conference (DASC), Prague, Czech Republic, 2015, pp. 2B3-1-2B3-10, doi: 10.1109/DASC.2015.7311363.

[12] T. Parcollet, M. Morchid and G. Linarès, "E2E-SINCNET: Toward Fully End-To-End Speech Recognition," ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 2020, pp. 7714-7718, doi: 10.1109/ICASSP40776.2020.9053954.

[13] S. Kim, T. Hori, and S. Watanabe, "Joint CTC-Attention based End-to-End Speech Recognition using Multi-task Learning," *arXiv.org*, 2016. https://arxiv.org/abs/1609.06773 (accessed Apr. 01, 2025).

[14] S. Schneider, A. Baevski, R. Collobert, and M. Auli, "wav2vec: Unsupervised Pre-training for Speech Recognition," *arXiv:1904.05862 [cs]*, Sep. 2019, Available: https://arxiv.org/abs/1904.05862

[15] J. Zuluaga-Gomez, P. Motlicek, Q. Zhan, K. Vesely, and R. Braun, "Automatic Speech Recognition Benchmark for Air-Traffic Communications," arXiv.org, 2020. http://arxiv.org/abs/2006.10304 (accessed Apr. 15, 2025).

[16] E. Pinska-Chauvin, H. Helmke, J. Dokic, P. Hartikainen, O. Ohneiser, and R. G. Lasheras, "Ensuring Safety for Artificial-Intelligence-Based Automatic Speech Recognition in Air Traffic Control Environment," Aerospace, vol. 10, no. 11, p. 941, Nov. 2023, doi: https://doi.org/10.3390/aerospace10110941.

[17] S. Badrinath and H. Balakrishnan, "Automatic Speech Recognition for Air Traffic Control Communications," Handle.net, Sep. 2022, doi: https://hdl.handle.net/1721.1/145275.

[18] A. Babu et al., "XLS-R: Self-supervised Cross-lingual Speech Representation Learning at Scale," arXiv.org, Dec. 16, 2021. https://arxiv.org/abs/2111.09296.

APPENDIX

1. Research Paper

AeroVoice: Automatic Speech Recognition for ATC communication

Dhruva Chaudhari
Department of Computer Engineering
Vivekanand Education Society's Institute
Of Technology (Affiliated to the
University of Mumbai)
Mumbai, India
2021.dhruva.chaudhari@ves.ac.in

Anurag Shirsekar
Department of Computer Engineering
Vivekanand Education Society's Institute
Of Technology (Affiliated to the
University of Mumbai)
Mumbai, India
2021.anurag.shirsekar@ves.ac.in

Preethika Shetty
Department of Computer Engineering
Vivekanand Education Society's Institute
Of Technology (Affiliated to the
University of Mumbai)
Mumbai, India
2021.preethika.shetty@ves.ac.in

Sneha Tanna
Department of Computer Engineering
Vivekanand Education Society's Institute
Of Technology (Affiliated to the
University of Mumbai)
Mumbai, India
2021.sneha.tanna@ves.ac.in

Lifna CS
Department of Computer Engineering
Vivekanand Education Society's
(Assistant Professor) Institute Of
Technology (Affiliated to the University
of Mumbai)
Mumbai, India
https://orcid.org/0000-0002-6054-2755

Abstract—**This paper introduces AeroVoice: an innovative system developed for applying ASR in Air Traffic Control communications. Conventional manual transcription and analytical techniques prove insufficient in dealing with the demands of real-time operations and have other drawbacks in light of the increasing complexity and volume of aviation exchanges. Due to Automatic Speech Recognition (ASR) and Natural Language Processing (NLP) in AeroVoice, ATC voice signals can now be transcribed and analysed with high accuracy and efficiency. Flight orders and callsigns, runway assignments and information about departures from normal operations can be identified and extracted by the system. The above processes are automated in AeroVoice, which enhances the awareness of the controllers and pilots, as well as reduces cognitive load and contributes to safety operations. The advanced algorithms are invoked in AeroVoice for different accents, noise levels, or pacing making it an effective and possible solution in the future of aviation voice communications. This paper explores AeroVoice's technical design, observes its efficiency in different working conditions, and outlines enhancements to enhance the system's accuracy and validity.**

Keywords—Air traffic Control (ATC), Automatic Speech Recognition (ASR), Speech-to-text, NLP

I. Introduction

Reliable communication between pilots and Air Traffic Controllers (ATCos) is critical for air safety. Traditional Voice over VHF channels, however, are susceptible to frequency congestion, miscommunication, and human error . To address these issues, Automatic Speech Recognition (ASR) systems have been increasingly adopted to automate and enhance air-ground communications .

Recent ASR advancements—especially with transformer-based models like wav2vec 2.0—have improved recognition accuracy in noisy ATC environments. Custom pipelines trained on ATC-specific datasets such as ATCO2 demonstrate the need for domain adaptation to handle unique linguistic patterns and acoustic challenges . These systems must also contend with code-switching, speaker variability, and rapid speech patterns, making ATC a particularly challenging domain for ASR.A key ASR application is callsign recognition, which supports radar label pre-filling and reduces ATCo workload. End-to-end approaches combining CTC and attention mechanisms have shown promise, though performance often degrades under real-world noise, necessitating further tuning .

As safety is paramount, research also focuses on AI risk assessment frameworks and the integration of secure communication systems like CPDLC . Emerging ASR tools—such as early callsign highlighting and semantic information extraction—are evolving into full-fledged Automatic Speech Understanding (ASU) systems. These advances not only improve operational efficiency but also contribute to incident prevention through proactive information extraction.

This project, *AeroVoice*, leverages these advances to build an ASR-ASU-based ATC Radio Conversation Analyzer that transcribes, understands, and structures ATC communications to assist controllers and improve operational safety.

II. Related Work

In the exploration of integrating Automatic Speech Recognition (ASR) into Air Traffic Control (ATC) systems, numerous helpful studies were identified.

The significant findings by Wang et al., who stressed the usefulness of corpora such as the Air Traffic Control Complete Corpus and the ATCSpeech Corpus for improving ASR accuracy in ATC communication. This research has led us to examine methods such as Gaussian Mixture methods (GMMs) and Deep Neural Networks (DNNs) for boosting performance in real-time ATC situations, despite the challenges provided by noise and language variety[1].

The ATCO2 project considerably improved the understanding of large-scale ASR dataset creation. The project provided 5,281 hours of ATC speech data to enhance Named Entity Recognition (NER) for callsigns and directives. This dataset helped shape data collection strategy and the development of robust transcription tools[2].

In building an effective ASR system, the SESAR2020 project provided useful lessons on real-time flight callsign recognition. Their approach achieved an excellent 84% controller accuracy rate, which inspired us to focus on increasing recognition in high-stress conditions[3].

Domain-shift adaptation in Automatic Speech Recognition (ASR) was investigated using Wav2Vec 2.0 models, demonstrating that self-supervised learning approaches can effectively adapt to Air Traffic Control (ATC) communications, even in challenging conditions such as noise and domain shifts[4].

ATCO2 serves as a large-scale resource to enhance both ASR and NLU processing systems in air traffic control domains through its 5,000-hour audio sample along with 4 hours of manual annotations. The ATCO2 corpus enables high task performance because it achieves F1 scores of 0.97 for callsign recognition and 0.82 for command identification. Further development remains necessary because background noise interferes with performance while the method of obtaining pseudo-labels requires improvement.[5].

The study by Yin, Lin, et al., used AI-driven speech communication in an ATC system, showing high trust (94.3%), and safety benefits (92.3%). The limitations were ethical concerns like bias[6].

A study also suggested the benefits of using Feature Learning, and end-to-end training with RNNs. It achieves a multilingual character error rate of 6.9%, outperforming baseline models. [7]

Shetty et al. demonstrated that ASR systems require surveillance data integration for effectively recognizing callsigns during their research. The researchers discovered that adding contextual information cut the identification mistakes in callsigns down to 2.8% for ATCos and to 4.5% for pilots. Researchers used the study findings as the basis to develop techniques for improving ASR models through enhanced metadata utilization for accuracy [8].

Air traffic communication depends on Very High Frequency (VHF) voice transmission yet this efficient method faces problems including misunderstandings and transmission quality decline together with frequency congestion because of increasing air traffic volumes. The pilots and Air Traffic Controllers must maintain continuous attention to verbal communication which leads to delays and non-safety-critical message inaccuracies due to readback/hearback errors as well as call sign misunderstandings and frequency variations. The communication flow gets disrupted most intensely when precision and clarity become crucial to operations.

Controller-Pilot Data Link Communication (CPDLC) became a solution for managing non-critical messages through text-based systems which cleared VHF channels for essential communications. CPDLC provides effective advantages through reduced frequency congestion alongside minimized routine communications misunderstandings yet it presents specific disadvantages. Sestorp and Lehto found in their CPDLC security evaluation that encryption used insufficiently while authentication procedures were inadequate and system integration processes were challenging to manage. The safety of ATC systems depends primarily on advanced encryption methods according to their findings while such encryption must receive regular security updates to protect against cyber intrusions. The identified system weak points increase doubts about message accuracy and security specifically when messages are sent too early or when the transmission environment is compromised. The reliance on human interaction during message logging and tracking combined with error correction causes CPDLC to maintain human error risks that prevent complete elimination of communication breakdowns.[9]

Kleinert et al. expanded an ontology for ATCo-pilot communication so the system could better interpret ATC instructions automatically. The model showed high success with extracting callsigns in 97.2% of cases from gold transcriptions yet achieved 90.3% accuracy from automated transcriptions which reveals a clear requirement for better ASR advancements for real-world ATC environments [10].

The research studies gave us important information to improve ATC communication while focusing on ASR accuracy as well as CPDLC security measures and automated ATCo-pilot exchange interpretation methods to guide the current investigation.

The table below shows the accuracy results obtained from various combinations of models used for performance evaluation.

Model	Year	Datasets Used	Methodology & Algorithms	Performance Metrics	Key Inferences / Limitations
GMM-HMM [1]	2024	LDC94S14A, NATO N4, ATCSpeech, LiveATC, ATCSC, AIRBUS-ATC	Statistical Models, GMMs, HMMs, TDNNs, Supervised/Unsupervised Learning	ConER: 50%, CmdER: 100%	Enhances ATC but may impair situational awareness, requires additional tools to catch ASR errors.
DNN-HMM Hybrid [2]	2023	ATCO2-T (5281h), ATCO2-test-set-1h, ATCO2-test-set-4h	Lattice Rescoring, Speaker Role Detection (SRD), Speaker Diarization (SD), End-to-End Neural Diarization	F1-Score: 60%-80%, NER Callsigns: 97%, Commands: 87.1%, WER: Improved with larger datasets	Handles noise & accents but struggles in challenging ATC conditions, resource-intensive fo real-time use.
End-to-End ASR [5]	2023	ATCO2 corpus (5000+ hours), 4h manually annotated	Hybrid ASR (HMMs + DNNs), E2E models, Acoustic Model, Feature Extraction (MFCCs, i-vectors), Speaker Diarization (BHMM, x-vectors), NER (BERT-based)	Speaker Role Detection: F1 0.86-0.83, ASR WER: 22.3%-11.1%	Requires extensive training data, struggles with noise, reliance on pseudo-labels may impact accuracy.

BILSTM + Hybrid ASR [3]	2023	1,000h ATC recordings (400h English, 600h Spanish)	Hybrid ASR, audio extraction, BILSTM for speech recognition, rule-based callsign detection	Controller Acc.: 84%, Flight Crew Acc.: 67%, Callsign request accuracy: 58.5%	Overlooks linguistic diversity, struggles with callsign phonetics, lacks noise handling.
Wav2Vec 2.0 [4]	2022	NATS (18h), ISAVIA (14h), ATCO2, LDC-ATCC, UWB-ATCC, ATCOSIM	Pre-trained self-supervised model, fine-tuned on domain-specific data	LDC-ATCC WER: 25.0%, UWB-ATCC WER: 54.6%, ATCO2 WER: 58.7%	Limited generalizability, overlooks demographic factors, requires robustness improvements.
Deep Speech + RNN [17]	2022	AIRBUS-ATC, ATCOSIM, NIST (LDC94S14A)	RNN with MFCCs, CTC loss, N-gram LM, Beam Search, Callsign Extraction	WER: 17%, N-gram improved accuracy by 26%, Callsign F1: 0.95	Struggles with ATC jargon, noise, accents limited extraction & regional generalization.
DNN + MIP for Risk Detection [6]	2022	Historical ATC recordings, system logs, operational data	ASR for air-ground communication, SIU for intent detection, DNN for risk detection, MIP & TSP for real-time analysis	Safety enhancement: 92.3%, Credibility 94.3%	Overlooks AI bias, lacks long-term impact assessment, limited ATCO feedback.
SincNet + CNN-RNN [7]	2021	ATCSpeech (Chinese, English, Multilingual), WSJ	SincNet for feature extraction, wav2vec for ASR, LSC for extraction, CNN-RNN	CER: Chinese: 7.6%, English: 8.9%, Multilingual: 6.9%	Effective but complex, needs simplification, requires better noise robustness.
TDNNF + Byte Pair Encoding [15]	2020	MALORCA, ATCOSIM, UWB ATCC, AIRBUS, ATCC USA, HIWIRE, Librispeech, Commonvoice	DNN, TDNN, CNN, TDNNF, Kaldi's Chain LF-MMI, N-gram LM	WER: 7.75%, 35% improvement with TDNNF & Byte-Pair Encoding	Struggles with accents, speaker biases, real-world deployment concerns.

Table 1: Comparison of various models for ASR

Observations from the Comparison Table:

Traditional vs. Deep Learning ASR:Because the GMM-HMM models work correctly with existing ATC systems they maintain a CmdER of 100%. The callsign recognition accuracy achieved 97% success while command detection scores 87.1% and the models demand big training sets and processing strength.

End-to-End ASR Needs Large Data but Shows High Accuracy:The implementation of WER reconstructed at 22.3% but was actually reduced to 11.1% along with Speaker Role Detection reaching an F1 score of 0.86-0.83. The system operates in noisy ATC conditions while using pseudo-labels that introduce possible errors to the analysis.

Hybrid ASR Models Improve Recognition but Have Limitations:BILSTM + Hybrid ASR demonstrates controller recognition at 84% but shows difficulty when identifying flight crew messages (67%) as well as callsign strings (58.5%). Phonetic variations together with linguistic diversity within ATC speech cause issues during transmissions.

Wav2Vec 2.0 Shows Potential but Lacks ATC-Specific Robustness:The ATC datasets show elevated WER statistics of 58.7% in ATCO2 and 54.6% in UWB-ATCC. The system encounters difficulties due to ATC terminology, various speech accents and noisy operational conditions.

Deep Speech + RNN & TDNNF Improve WER but Have Deployment Issues:Deep Speech + RNN reaches a WER equal to 17% yet N-gram modeling enhances its accuracy by 26%. The WER performance of TDNNF decreased by 35% yet the system demonstrates limitations when processing accents and speakers in natural operating environments.

CNN-Based & Risk-Detection Models Improve Safety but Are Complex:The CER performance of SincNet + CNN-RNN amounts to 7.6% Chinese and 8.9% English making it suitable for multilingual ATC operations but challenging to implement. The combination of DNN + MIP increases ATC safety levels to 92.3% however it does not provide assessment of sustained impact or receive feedback from ATCOs.

Summary: Deep learning models improve accuracy but struggle with real-time ATC deployment due to noise, accents, and computational challenges. Future work should focus on hybrid models with noise reduction and domain adaptation techniques.

III. Proposed Methodology

Figure 1 illustrates the system architecture of *AeroVoice*, a framework designed to process and analyze air traffic control (ATC) communications. The workflow begins with recorded ATC audio, either captured from live radio transmissions or other sources, which undergoes audio preprocessing. The preprocessed audio is then fed into the Automatic Speech Recognition (ASR) module for transcription and the Natural Language Processing (NLP) module for extracting critical information. Both modules interact with the input and output components to generate transcriptions and structured data, facilitating downstream tasks like call sign identification and deviation detection.

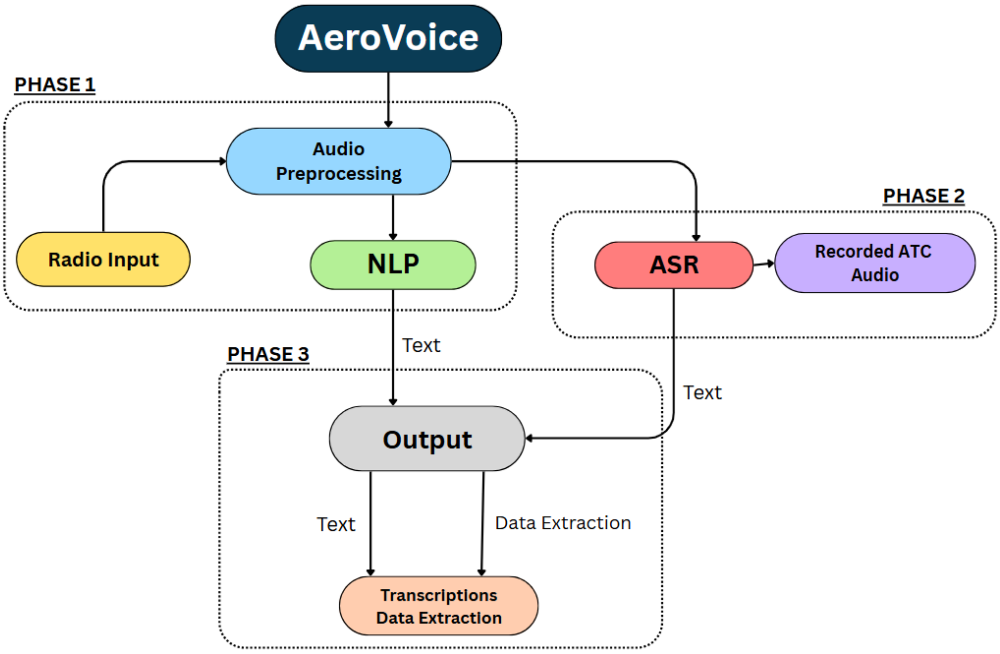


Figure 1: ASR System Architecture for ATC Communication

A. System Architecture

ASR Pipeline based on Radio Inputs (Phase 1)

The ASR pipeline serves as the initial stage of AeroVoice to transform unprocessed ATC audio into written transcription text. The ASR model must maintain accurate performance in ATC communications since controllers issue commands followed by pilot response confirmations but must face difficulties with background noise along with overlapping speech and variable speech speeds.

Input: The pipeline operates on raw ATC audio recordings that stem from active radio transmissions together with pre-recorded sources. The audio files consist of aviation-specialized words combined with numerical flight orders while including call signs and need complete transcription before analysis.

Process: The ASR model combined with Wav2Vec processing provides specialist performance in recognizing ATC terminology and altitudes and headings and command statements. The system improves its operational speed by dividing input audio into five to ten second sections to achieve both quick processing and relevant information preservation.

The processes carried out by the ASR pipeline function as follows:

1. Audio Segmentation function enables the system to split lengthy recordings into multiple sections thus increasing both efficiency and accuracy of transcription results.

2. The speech feature processing procedure applies to segmented audio files by prioritizing numerical information and typical air traffic control statements.

3. The system transforms obtained features into textual content while maintaining both proper formatting and ATS phrase compliance.

Output: A structured ATC communication transcription emerges from the process to act as an input for detecting errors after recognizing call signs while performing real-time monitoring. The ASR pipeline's accuracy level determines how well downstream processes operate thus making it an essential part of AeroVoice operation.

Automatic Speech Understanding (ASU) (Phase 2)

ATC communications are rewritten into text by the Automatic Speech Understanding (ASU) module so that the system can obtain essential entities while recognizing communication errors and spotting emergency conditions and spotting flight parameter inconsistencies. The extracted data gets organized during this phase into meaningful information that matches the aviation domain which facilitates better monitoring decisions.

The module applies entity extraction to mark down vital components in the transcribed material. The identified entities consist of altitude, heading, speed together with waypoints, flight numbers, call signs and command phrases. This extraction methodology allows future operations of error detection and intent classification and emergency recognition systems. Entity extraction plays a vital role to guarantee correct recording and analysis of flight parameters.

Under the next step the system begins intent classification by categorizing commands issued by ATC based on their specified purpose. The different types of common intent fall into categories such as altitude assignments, heading changes, clearance approvals, handovers and emergency declarations. The classification of communication intentions enables us to verify that the messages follow standard Air Traffic Control processes.

The system conducts error detection together with conflict analysis by matching extracted flight parameters with predefined values. The system detects height differences between reported altitudes and issued commands together with unaccepted flight instructions and dual airspace clearance points and route path discrepancies. When system detection identifies an inconsistency it sends the matter forward for human examination.

The ASU module includes built-in emergency detection systems which scan for emergency signals in radio transmissions. The system instantly detects explicit emergency signals including “Mayday” and “Pan-Pan” but uses additional assessment tools to handle indirect emergency signs that involve unresponsive pilot conduct.

The ASU module unifies entity extraction alongside intent classification together with error detection and emergency identification to supply an organized and extensive interpretation of ATC information streams. The processed information serves as input for successive phases that promote both monitoring capabilities and visual enhancement which results in better air traffic safety and efficiency.

Real-Time Web Interface Algorithm (Phase 3)

The system provides easy web interface access to ATCos who need to view transcriptions along with detected issues during this processing phase. The FastAPI backend retrieves processed transcriptions with flagged issues from the backend for display through the React-based front end in an organized interface.

The system presents updated information automatically as new transcription content becomes available so ATCos can easily track flagged communication errors and check communication logs upon demand. The system facilitates fast error detection which results in better ATC communication evaluation performance.

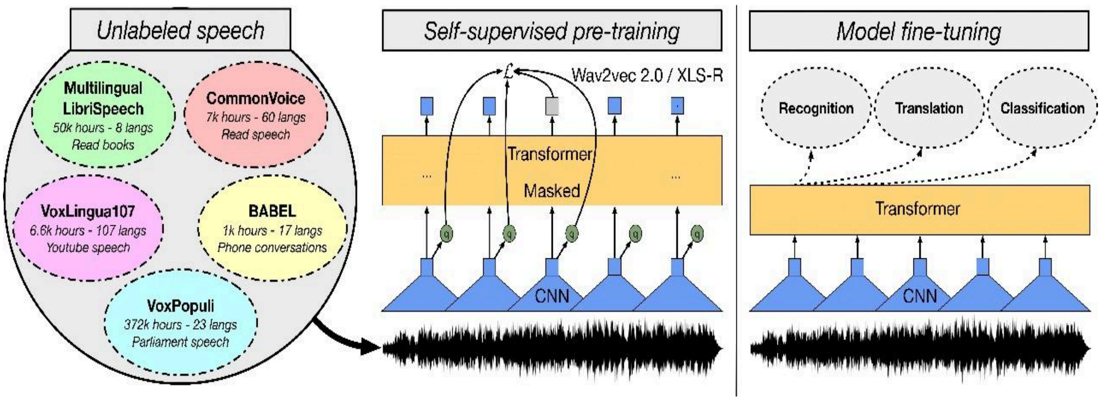


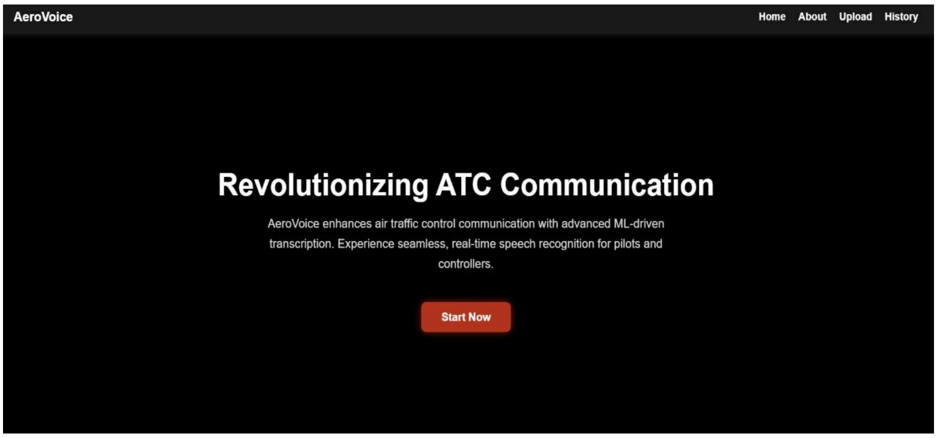
Figure 2: Architecture of wav2vec model[18]

As shown in Figure 2, the Wav2Vec 2.0 / XLS-R model in AeroVoice uses unlabeled multilingual data, with CNN-Transformer pre-training followed by fine-tuning for tasks like speech recognition, enabling strong performance on noisy, accented ATC speech

IV. Implementation & Result

4.1 Technical Stack

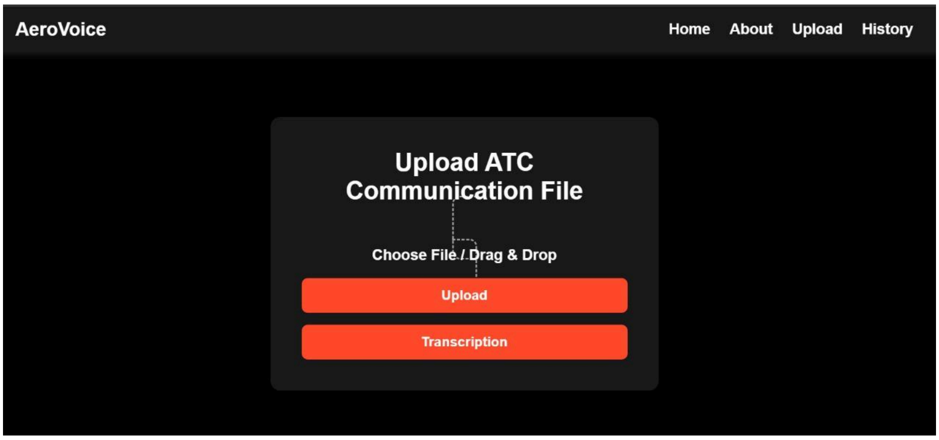
AeroVoice ASR employs machine learning along with natural language processing and web technologies to transform ATC audio into precise transcriptions through an error detection feature which also enables easy communication monitoring.



Revolutionizing ATC Communication

AeroVoice enhances air traffic control communication with advanced ML-driven transcription. Experience seamless, real-time speech recognition for pilots and controllers.

Figure 3 : Screenshot of Home Page

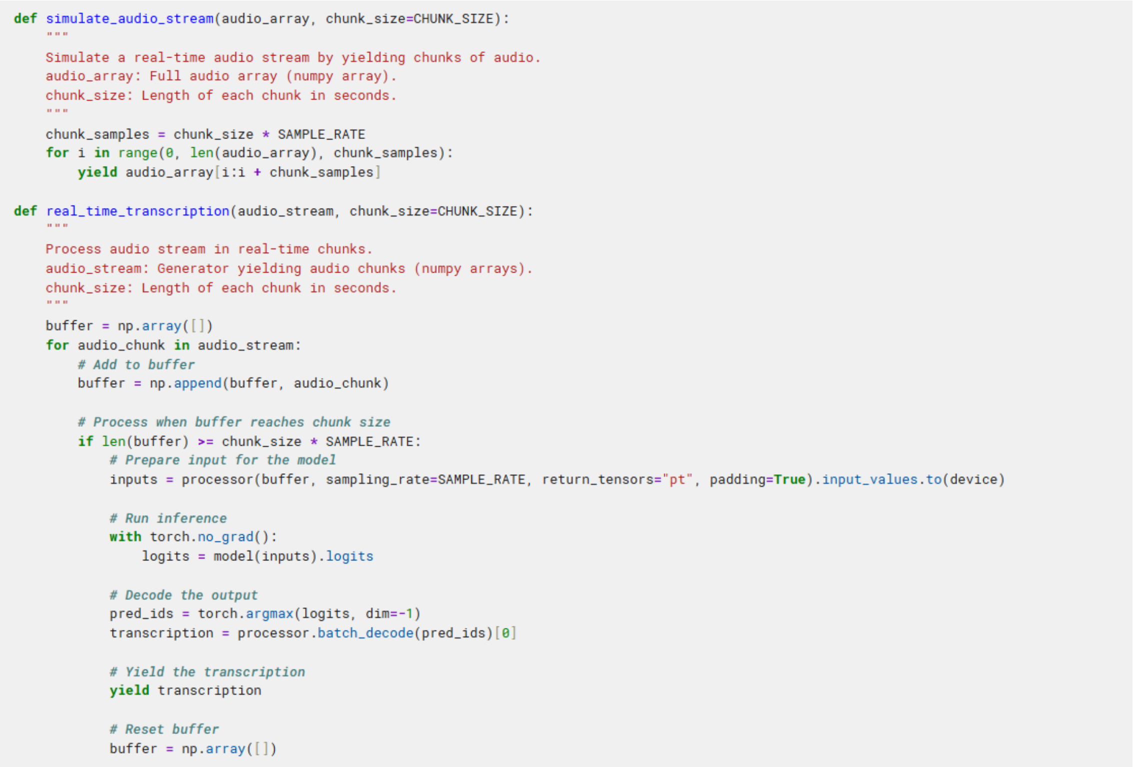


Upload ATC Communication File

Figure 4: Screenshot of Upload Page

4.2 Automatic Speech Recognition (ASR)

The ASR component employs Wav2Vec 2.0 from PyTorch to convert ATC audio into written text. The system receives aviation-specific datasets for training that help it identify ATC terminology more effectively. Deep learning and speech processing functions in the system rely on torch and torchaudio libraries together with Transformers for accessing Wav2Vec 2.0 models from Hugging Face and speechbrain as an open-source speech processing toolkit. The ATC processing system begins by accepting audio files then uses Wav2Vec 2.0 features extraction to create text output while performing simple formatting changes for the final result.



```
def simulate_audio_stream(audio_array, chunk_size=CHUNK_SIZE):
    """
    Simulate a real-time audio stream by yielding chunks of audio.
    audio_array: Full audio array (numpy array).
    chunk_size: Length of each chunk in seconds.
    """
    chunk_samples = chunk_size * SAMPLE_RATE
    for i in range(0, len(audio_array), chunk_samples):
        yield audio_array[i:i + chunk_samples]

def real_time_transcription(audio_stream, chunk_size=CHUNK_SIZE):
    """
    Process audio stream in real-time chunks.
    audio_stream: Generator yielding audio chunks (numpy arrays).
    chunk_size: Length of each chunk in seconds.
    """
    buffer = np.array([])
    for audio_chunk in audio_stream:
        # Add to buffer
        buffer = np.append(buffer, audio_chunk)

        # Process when buffer reaches chunk size
        if len(buffer) >= chunk_size * SAMPLE_RATE:
            # Prepare input for the model
            inputs = processor(buffer, sampling_rate=SAMPLE_RATE, return_tensors="pt", padding=True).input_values.to(device)

            # Run inference
            with torch.no_grad():
                logits = model(inputs).logits

            # Decode the output
            pred_ids = torch.argmax(logits, dim=-1)
            transcription = processor.batch_decode(pred_ids)[0]

            # Yield the transcription
            yield transcription

            # Reset buffer
            buffer = np.array([])
```

Figure 5:Screenshot of code snippet for Automatic Speech Recognition

4.3 Audio Preprocessing

The system performs multiple preprocessing operations as part of audio optimization to ensure accurate transcription results. The system applies two processing methods to audio by resampling to standard 16 kHz frequency then dividing extended ATC recordings into segments between 5-10 seconds. The system provides capabilities to convert audio files into compatible formats for processing. This preprocessing relies on libraries such as librosa for audio signal processing and pydub for segmentation and format conversion.

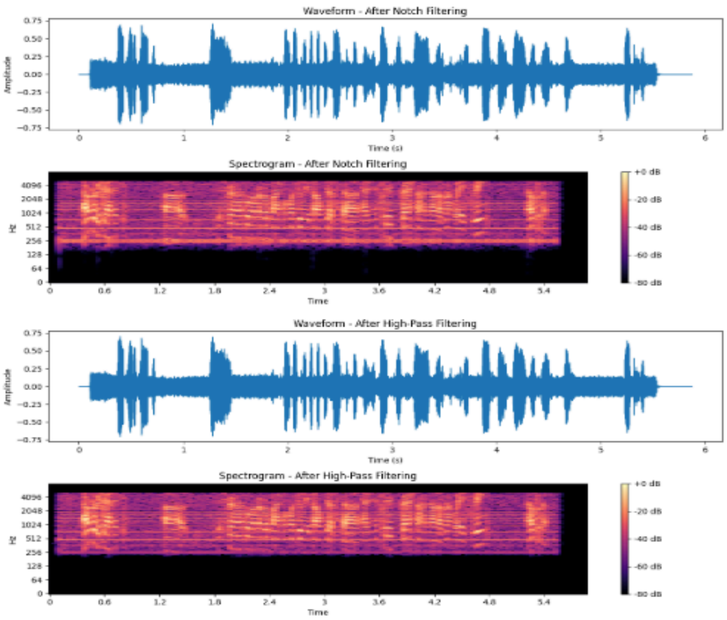
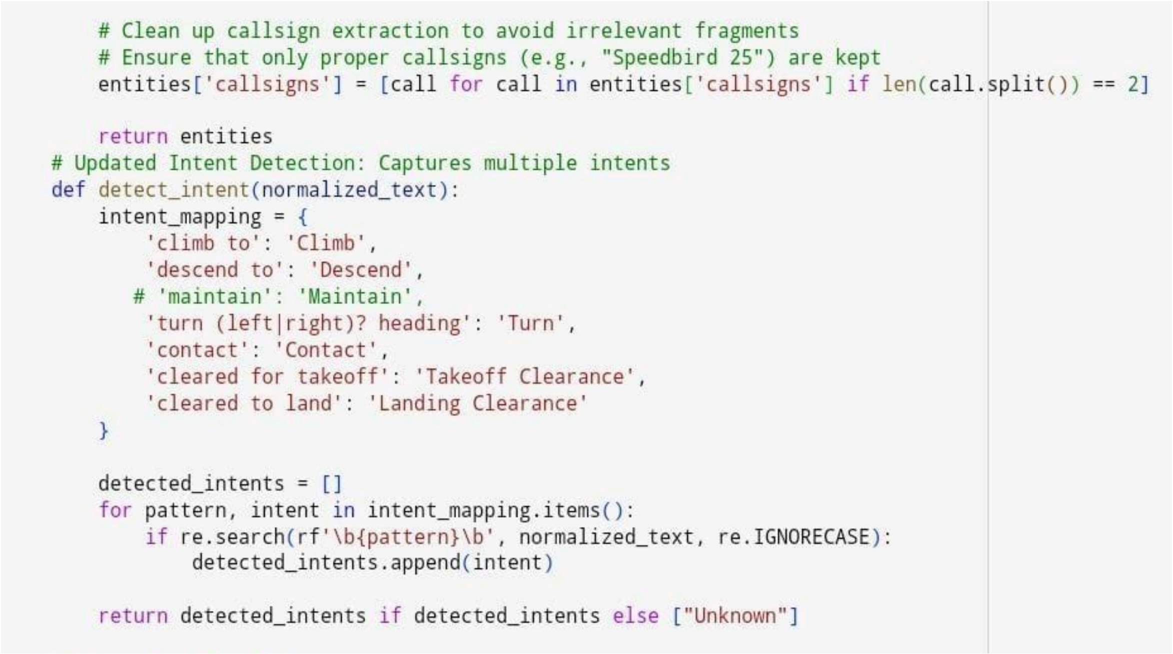


Figure 6: Screenshot of Audio Preprocessing using techniques - Notch Filtering , High Pass Filtering

4.4 Intent Detection & Flagging

The system evaluates ATC commands together with pilot responses to detect communication mistakes through established rules. This functionality employs Regular Expressions for pattern detection in ATC crew communications, NLTK for tokenization and keyword extraction, and pandas for data handling during the intent detection process.

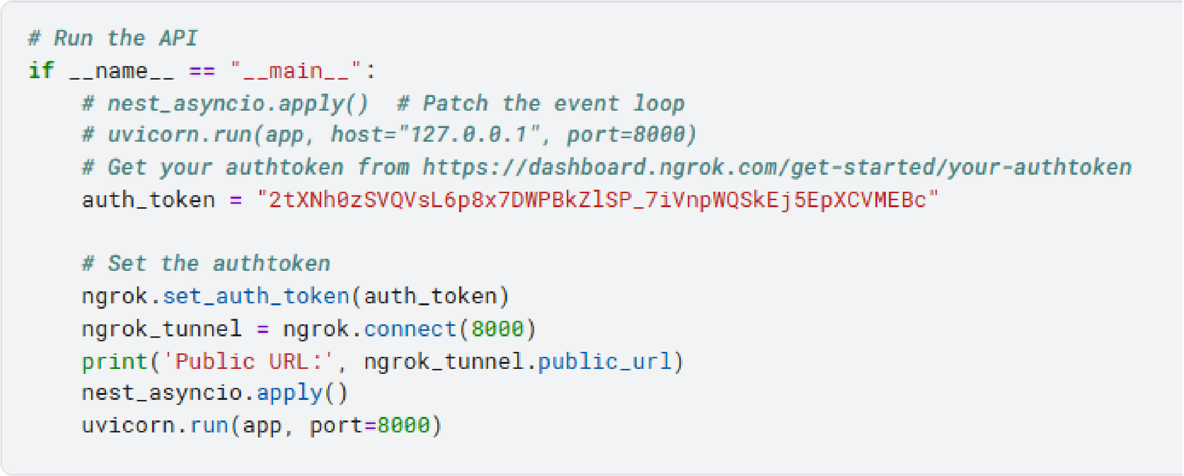


```
    # Clean up callsign extraction to avoid irrelevant fragments
    # Ensure that only proper callsigns (e.g., "Speedbird 25") are kept
    entities['callsigns'] = [call for call in entities['callsigns'] if len(call.split()) == 2]

    return entities
# Updated Intent Detection: Captures multiple intents
def detect_intent(normalized_text):
    intent_mapping = {
        'climb to': 'Climb',
        'descend to': 'Descend',
       # 'maintain': 'Maintain',
        'turn (left|right)? heading': 'Turn',
        'contact': 'Contact',
        'cleared for takeoff': 'Takeoff Clearance',
        'cleared to land': 'Landing Clearance'
    }

    detected_intents = []
    for pattern, intent in intent_mapping.items():
        if re.search(rf'\b{pattern}\b', normalized_text, re.IGNORECASE):
            detected_intents.append(intent)

    return detected_intents if detected_intents else ["Unknown"]
```

Figure 7: Screenshot of code snippet for intent detection

4.5 Backend System

The backend system depends on FastAPI to handle transcription requests and execute ASR computations that produce monitored error flags. The API functionality relies on fastapi while the backend server runs uvicorn for FastAPI applications plus the backend uses pydantic for data structure validation within communication responses. The backend workflow begins with inputted ATC audio which is processed through ASR transcription before error detection for final responses to the frontend.



```
# Run the API
if __name__ == "__main__":
    # nest_asyncio.apply()  # Patch the event loop
    # uvicorn.run(app, host="127.0.0.1", port=8000)
    # Get your authtoken from https://dashboard.ngrok.com/get-started/your-authtoken
    auth_token = "2tXNh0zSVQVsL6p8x7DWPBkZ1SP_7iVnpWQSkEj5EpXCVMEBc"

    # Set the authtoken
    ngrok.set_auth_token(auth_token)
    ngrok_tunnel = ngrok.connect(8000)
    print('Public URL:', ngrok_tunnel.public_url)
    nest_asyncio.apply()
    uvicorn.run(app, port=8000)
```

Figure 8: Screenshot of code snippet for FastAPI

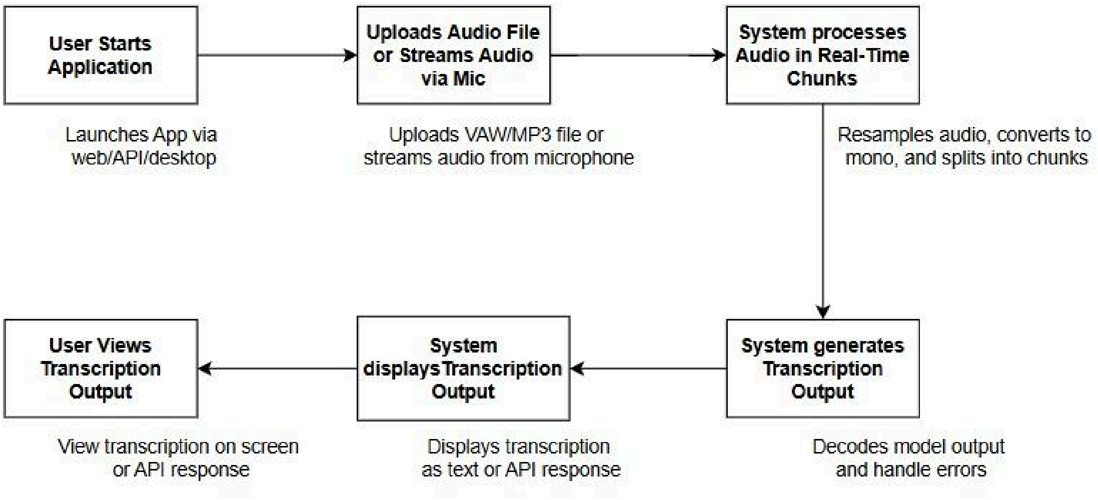

User Starts Application
Launches App via web/API/desktop
Uploads Audio File or Streams Audio via Mic
Uploads VAW/MP3 file or streams audio from microphone
System processes Audio in Real-Time Chunks
Resamples audio, converts to mono, and splits into chunks
System generates Transcription Output
Decodes model output and handle errors
System displays Transcription Output
Displays transcription as text or API response
User Views Transcription Output
View transcription on screen or API response

Figure 9: User flow diagram of the ASR system.

Figure 3 demonstrates its user workflow of the Automatic Speech Recognition (ASR) system which tracks the user application interaction. When users start their interaction by launching the application they can proceed to either upload or stream audio. Following real-time audio processing the system divides the audio into proper sections to produce transcription output. The system shows the transcribed text to the user as a last step.

4.6 Performance and evaluation metrics of models

The visual representation found in Figure 3 demonstrates the WER evaluation of these models so users can understand their performance distinction better.

The bar chart presents WER values of seven ASR models starting from GMM-HMM through conventional methods to modern deep learning systems which include Wav2Vec 2.0. The data presented in the visual document indicates:

- The recognition performance of traditional models such as GMM-HMM remains low due to their elevated WER.
- The combination of neural networks in DNN-HMM models leads to moderate WER reduction because of their improved capability.
- The end-to-end models including DeepSpeech and Wav2Vec 2.0 achieve superior performance compared to other methods because of their neural network capabilities for processing sophisticated speech patterns to decrease errors significantly.

Self-supervised Wav2Vec 2.0 achieves the best performance among the tested models based on WER results indicating its superiority for modern ASR tasks. A visual depiction validates the table results by demonstrating how ASR technology now uses deep learning methods with self-supervision to achieve better performance.

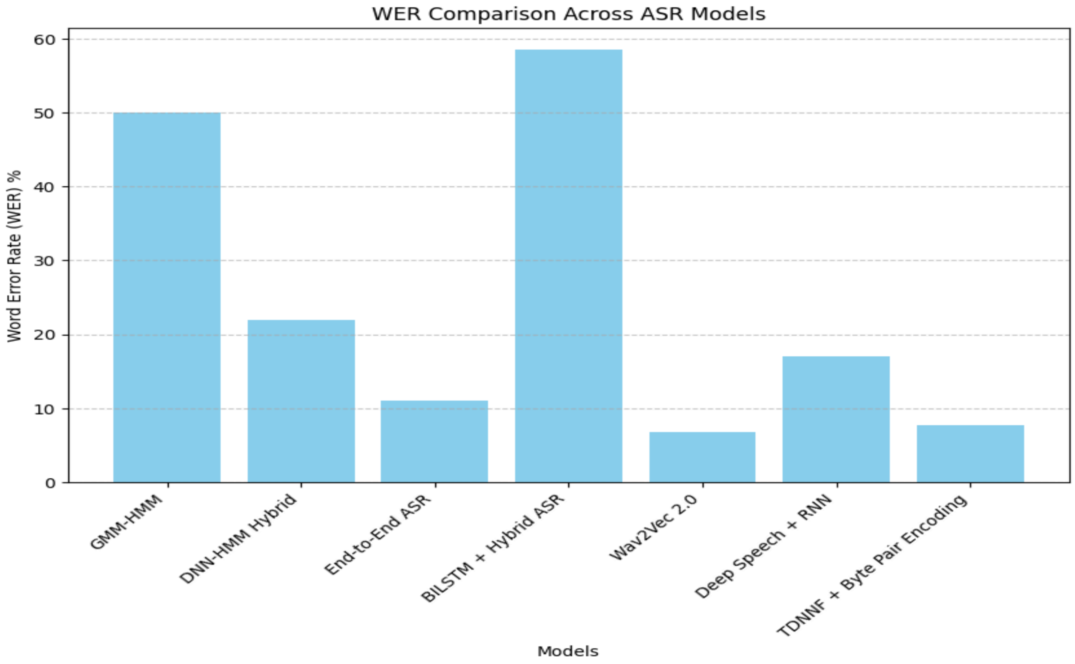


Figure 10: WER comparison across ASR models

V. Conclusion

The AeroVoice project successfully showcases how the combination of ASR and NLP technologies enhances ATC communication efficiency operations. Through the investigation of many ASR models we established the main advantages and drawbacks of traditional systems and deep learning-based approaches when applied in ATC environments. Real-time deployment of deep learning models faces obstacles because of their limitation with dealing with noise interference and computational overhead and the variations in accents in the system.

Future efforts will focus on reducing latency through optimized streaming architectures, improving noise handling with advanced filtering techniques, and developing hybrid models tailored for ATC-specific terminology and variations. The system will benefit from powerful logging methods coupled with artificial intelligence algorithms for improving transcription accuracy and generating valuable information for air traffic control officers.

The solutions implemented at AeroVoice close the distance between speech recognition improvements and operational ATC implementation. Continuous optimization of this system and real-time deployment capabilities will produce better air traffic management results by cutting manual workloads while boosting situational awareness and improving communication accuracy throughout aviation operations.

VI. Future Work

AeroVoice will get future updates through real-time ATC communication by eliminating delays and delivering quick transcriptions that require little latency. The system will use advanced noise reduction capabilities to eliminate background sounds and enhance difficult ATC voice signals. The system will receive sped-up optimization that shortens speech recognition processing duration to produce almost instantaneous transcription results. The system will establish a comprehensive logging solution which tracks flights through real-time monitoring while automatically finding errors in communication together with saving potential analysis material. AeroVoice transforms into a faster more accurate system along with improving reliability when used for ATC operations.

References

[1] Wang Z, Jiang P, Wang Z, Han B, Liang H, Ai Y, Pan W. Enhancing Air Traffic Control Communication Systems with Integrated Automatic Speech Recognition: Models, Applications and Performance Evaluation. *Sensors*. 2024; 24(14):4715. https://doi.org/10.3390/s24144715

[2] Zuluaga-Gomez J, Nigmatulina I, Prasad A, Motlicek P, Khalil D, Madikeri S, Tart A, Szoke I, Lenders V, Rigault M, et al. Lessons Learned in Transcribing 5000 h of Air Traffic Control Communications for Robust Automatic Speech Understanding. *Aerospace*. 2023; 10(10):898. https://doi.org/10.3390/aerospace10100898

[3] García R, Albarrán J, Fabio A, Celorrio F, Pinto de Oliveira C, Bárcena C. Automatic Flight Callsign Identification on a Controller Working Position: Real-Time Simulation and Analysis of Operational Recordings. *Aerospace*. 2023; 10(5):433. https://doi.org/10.3390/aerospace10050433

[4] Prasad, A., Nigmatulina, I., Sarfjoo, S., Motlicek, P., Kleinert, M., Helmke, H., Ohneiser, O., & Zhan, Q. (2022). How Does Pre-trained Wav2Vec 2.0 Perform on Domain Shifted ASR? An Extensive Benchmark on Air Traffic Control Communications. *ArXiv*. https://arxiv.org/abs/2203.16822

[5] J. Zuluaga-Gomez *et al.*, “ATCO2 corpus: A Large-Scale Dataset for Research on Automatic Speech Recognition and Natural Language Understanding of Air Traffic Control Communications,” *arXiv.org*, 2022. https://arxiv.org/abs/2211.04054 (accessed Apr. 01, 2025).

[6] Lin, Yi & Ruan, Min & Cai, Kunjie & Li, Dan & Zeng, Ziqiang & Li, Fan & Yang, Bo. (2022).“Identifying and managing risks of AI-driven operations: A case study of automatic speech recognition for improving air traffic safety,” Chinese Journal of Aeronautics, Aug. 2022, doi: https://doi.org/10.1016/j.cja.2022.08.020.

[7] P. Fan, D. Guo, Y. Lin, B. Yang, and J. Zhang, “Speech recognition for air traffic control via feature learning and end-to-end training,” arXiv.org, 2021. https://arxiv.org/abs/2111.02654

[8] S. Shetty, H. Helmke, M. Kleinert, and O. Ohneiser, “Early Callsign Highlighting using Automatic Speech Recognition to Reduce Air Traffic Controller Workload,” *AHFE international*, Jan. 2022, doi: https://doi.org/10.54941/ahfe1002493.

[9] I. Sestorp and A. Lehto, “CPDLC in Practice : A Dissection of the Controller Pilot Data Link Communication Security,” Dissertation, 2019.

[10] M. Kleinert et al., "Automated Interpretation of Air Traffic Control Communication: The Journey from Spoken Words to a Deeper Understanding of the Meaning," 2021 IEEE/AIAA 40th Digital Avionics Systems Conference (DASC), San Antonio, TX, USA, 2021, pp. 1-9, doi: 10.1109/DASC52595.2021.9594387

[11] H. Glaser-Opitz and L. Glaser-Opitz, "Evaluation of CPDLC and voice communication during approach phase," 2015 IEEE/AIAA 34th Digital Avionics Systems Conference (DASC), Prague, Czech Republic, 2015, pp. 2B3-1-2B3-10, doi: 10.1109/DASC.2015.7311363.

[12] T. Parcollet, M. Morchid and G. Linarès, "E2E-SINCNET: Toward Fully End-To-End Speech Recognition," ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 2020, pp. 7714-7718, doi: 10.1109/ICASSP40776.2020.9053954.

[13] S. Kim, T. Hori, and S. Watanabe, “Joint CTC-Attention based End-to-End Speech Recognition using Multi-task Learning,” *arXiv.org*, 2016. https://arxiv.org/abs/1609.06773 (accessed Apr. 01, 2025).

[14] S. Schneider, A. Baevski, R. Collobert, and M. Auli, “wav2vec: Unsupervised Pre-training for Speech Recognition,” *arXiv:1904.05862 [cs]*, Sep. 2019, Available: https://arxiv.org/abs/1904.05862

[15] J. Zuluaga-Gomez, P. Motlicek, Q. Zhan, K. Vesely, and R. Braun, “Automatic Speech Recognition Benchmark for Air-Traffic Communications,” arXiv.org, 2020. http://arxiv.org/abs/2006.10304 (accessed Apr. 15, 2025).

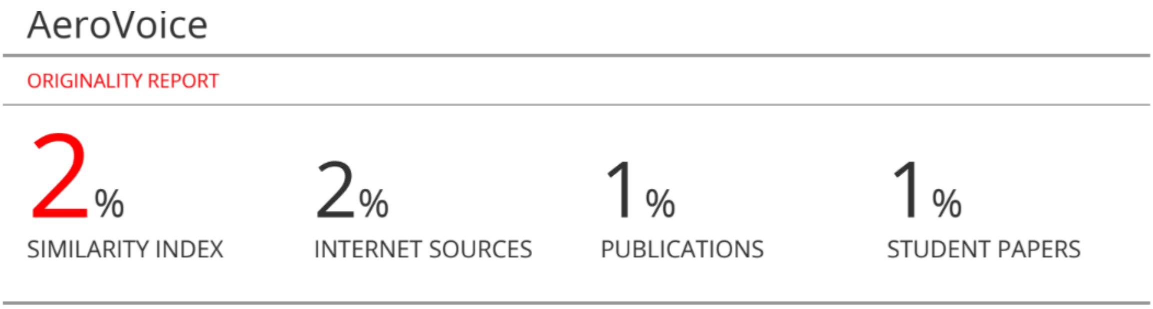
[16] E. Pinska-Chauvin, H. Helmke, J. Dokic, P. Hartikainen, O. Ohneiser, and R. G. Lasheras, “Ensuring Safety for Artificial-Intelligence-Based Automatic Speech Recognition in Air Traffic Control Environment,” Aerospace, vol. 10, no. 11, p. 941, Nov. 2023, doi: https://doi.org/10.3390/aerospace10110941.

[17] S. Badrinath and H. Balakrishnan, “Automatic Speech Recognition for Air Traffic Control Communications,” Handle.net, Sep. 2022, doi: https://hdl.handle.net/1721.1/145275.

[18] A. Babu et al., “XLS-R: Self-supervised Cross-lingual Speech Representation Learning at Scale,” arXiv.org, Dec. 16, 2021. https://arxiv.org/abs/2111.09296.

2. Paper Details

a. Plagiarism report



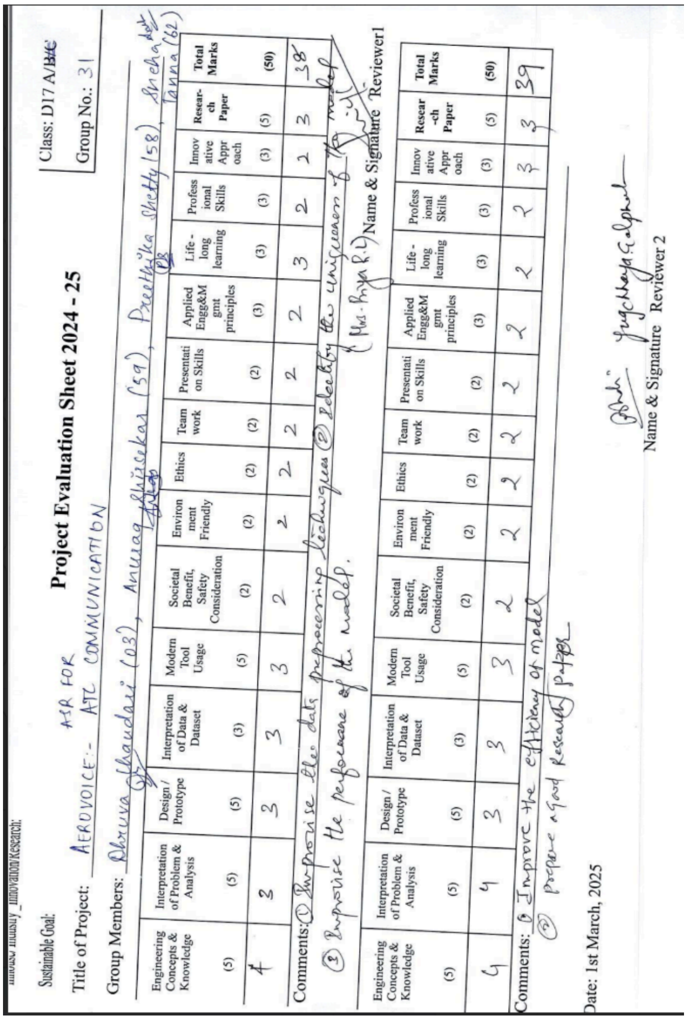
AeroVoice

ORIGINALITY REPORT

2%	2%	1%	1%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

b. Project review sheet

i. Review 1 (1st March, 2025)



Inhouse/ Industry_Innovation/Research:

Sustainable Goal:

Project Evaluation Sheet 2024 - 25

Class: D17 A/~~B/C~~

Group No.: 31

Title of Project: AERO VOICE :- ASR FOR ATC COMMUNICATION

Group Members: Dhruva Chaudhari (03), Anurag Shirsekar (59), Preethika Shetty (58), Sneha Tanna (62)

Engineering Concepts & Knowledge (5)	Interpretation of Problem & Analysis (5)	Design / Prototype (5)	Interpretation of Data & Dataset (3)	Modern Tool Usage (5)	Societal Benefit, Safety Consideration (2)	Environ ment Friendly (2)	Ethics (2)	Team work (2)	Presentati on Skills (2)	Applied Engg&M gmt principles (3)	Life - long learning (3)	Profess ional Skills (3)	Innov ative Appr oach (3)	Resear-ch Paper (5)	Total Marks (50)
4	3	3	3	3	2	2	2	2	2	2	3	2	2	3	38

Comments: ① Improvise the data preprocessing techniques ② Identify the uniqueness of the model ③ Improvise the performance of the model.

(Mrs. Priya R.L) Name & Signature Reviewer1

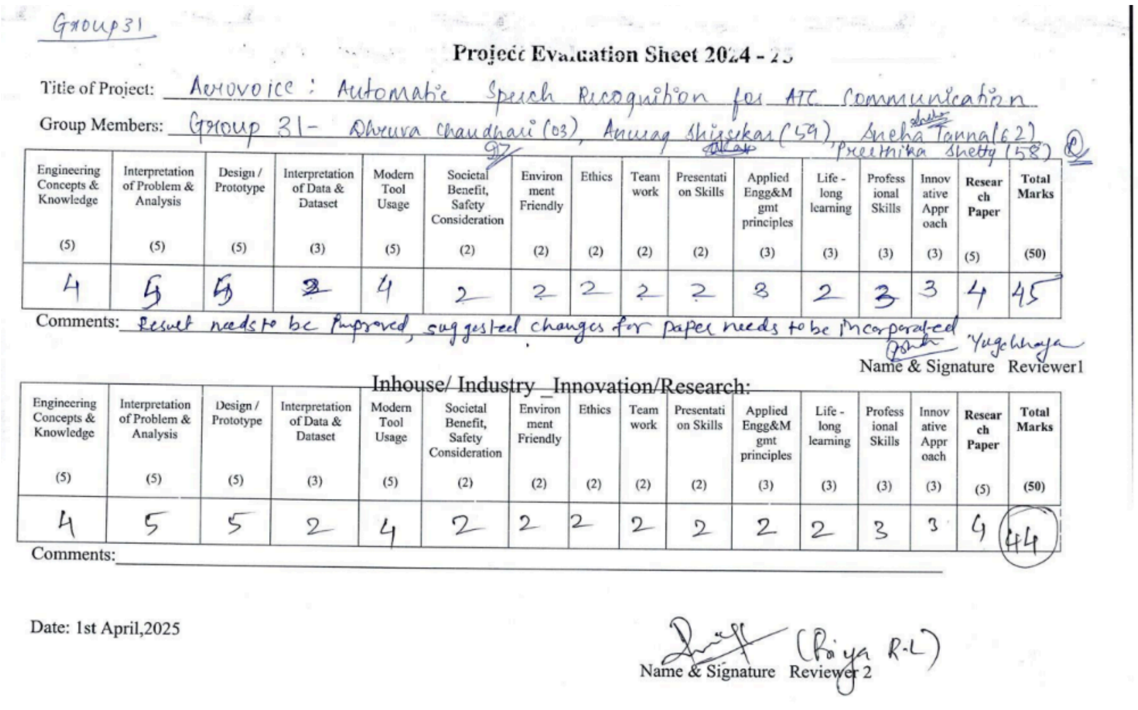
Engineering Concepts & Knowledge (5)	Interpretation of Problem & Analysis (5)	Design / Prototype (5)	Interpretation of Data & Dataset (3)	Modern Tool Usage (5)	Societal Benefit, Safety Consideration (2)	Environ ment Friendly (2)	Ethics (2)	Team work (2)	Presentati on Skills (2)	Applied Engg&M gmt principles (3)	Life - long learning (3)	Profess ional Skills (3)	Innov ative Appr oach (3)	Resear -ch Paper (5)	Total Marks (50)
4	4	3	3	3	2	2	2	2	2	2	2	2	3	3	39

Comments: ① Improve the efficiency of model ② prepare a Good Research paper

Date: 1st March, 2025

Yugchhaya Galphade
Name & Signature Reviewer 2

ii. Review 2 (1st April, 2025)



Group31

Project Evaluation Sheet 2024 - 2

Title of Project: Aerovoice : Automatic Speech Recognition for ATC communication

Group Members: Group 31- Dhruva Chaudhari (03), Anurag Shirsekar (59), Sneha Tanna (62), Preethika Shetty (58)

Engineering Concepts & Knowledge	Interpretation of Problem & Analysis	Design / Prototype	Interpretation of Data & Dataset	Modern Tool Usage	Societal Benefit, Safety Consideration	Environ ment Friendly	Ethics	Team work	Presentati on Skills	Applied Engg&M gmt principles	Life - long learning	Profess ional Skills	Innov ative Appr oach	Resear ch Paper	Total Marks
(5)	(5)	(5)	(3)	(5)	(2)	(2)	(2)	(2)	(2)	(3)	(3)	(3)	(3)	(5)	(50)
4	5	5	2	4	2	2	2	2	2	3	2	3	3	4	45

Comments: Result needs to be improved, suggested changes for paper needs to be incorporated

Yugchhaya
Name & Signature Reviewer1

Inhouse/ Industry _Innovation/Research:

Engineering Concepts & Knowledge	Interpretation of Problem & Analysis	Design / Prototype	Interpretation of Data & Dataset	Modern Tool Usage	Societal Benefit, Safety Consideration	Environ ment Friendly	Ethics	Team work	Presentati on Skills	Applied Engg&M gmt principles	Life - long learning	Profess ional Skills	Innov ative Appr oach	Resear ch Paper	Total Marks
(5)	(5)	(5)	(3)	(5)	(2)	(2)	(2)	(2)	(2)	(3)	(3)	(3)	(3)	(5)	(50)
4	5	5	2	4	2	2	2	2	2	2	2	3	3	4	44

Comments:

Date: 1st April,2025

(Priya R-L)
Name & Signature Reviewer 2