PROJECT REPORT

Audio-based Criminal Lie Detection System

Group 13

Group Members

Liu Mohan (Leader)

Zhang Zhiyuan

Zhong Wenyu

Zhou Yuhao

Zhang Lixin

Submit Date:

Oct 27, 2024

目录

1. Executive Summary

Our project develops an advanced speech recognition-based polygraph system with the aim of providing an innovative alternative to the traditional polygraph for criminal investigations and other areas. The traditional methods, for example the polygraph, have been the subject of criticism on ethical grounds and with regard to their reliability. In view of this, we propose a voice-based polygraph model that uses machine learning to enhance detection accuracy. This provides a more transparent, non-intrusive and adaptable solution for detection.

The project addresses the shortcomings of existing polygraph techniques, particularly in regard to data transparency, model interpretability and cross-domain applicability. They are achieved by integrating a range of machine learning models (including Random Forests, Support Vector Machines, KNN and others) and we utilise the soft-voting integration methods to enhance the reliability and accuracy of the predictions.

From a commercial perspective, the project's voice lie detector system has the potential for a wide range of applications in multiple fields, including criminal justice, corporate censorship and insurance claims. Our system can be offered on a per-use or subscription basis through a software-as-a-service (SaaS) cloud platform model, making it suitable for a variety of users, including law enforcement agencies, healthcare organisations, insurance companies, and corporate users. The system has been developed with the objective of meeting the specific needs of a range of enterprises. It could assist customers in making efficient judgments in different scenarios such as employee selection, internal vetting, and fraud detection.

2. Introduction

2.1 Project Background

Lying is common in human and non-human primate interactions, and is often divided into harmless altruistic falsehoods and serious denials of criminal activity. The focus of lie detection research has been on identifying the latter, where the liar often restrain emotional cues such as guilt and fear, making detection difficult[1]. This pervasiveness of deceptive behaviour has led many researchers to seek more scientific and accurate methods of lie detection. Given its importance in high-stakes areas, improving the accuracy of lie detection technologies is crucial.

In recent years, despite advances in lie detection, effective training and assessment materials remain a challenge, particularly when it comes to determining the truthfulness of a statement. Traditional polygraphs rely heavily on experts who undergo costly training for manual interrogation, and there are significant physiological responses that can be controlled by subjects with sufficient training or calmness, leading to unreliable results. And traditional simulations, which simply instruct participants to lie, fail to replicate the complexity of real deception scenarios[2]. While traditional polygraph tests remain useful, they have been criticised for their reliability and ethical implications. These limitations have led researchers to develop artificial environments that more accurately simulate real-life conditions. As a result, the integration of up-to-date artificial intelligence algorithms, computational power and domain knowledge into lie detection is becoming increasingly important as it offers the potential to improve the effectiveness of these simulations[3].

In the area of criminal investigation, as noted by criminologist Inbau, physical evidence alone proves insufficient for convicting suspects, and, breakthroughs typically arise from verbal testimonies and confessions during trials[4]. This underscores the importance of reliable methods to authenticate spoken evidence, inspiring the solution provided by voice recognition technologies as an alternative to conventional methods that are often intrusive

However, the effectiveness of speech recognition systems in lie detection is still a subject of ongoing research. However, the potential of voice recognition technology to improve criminal investigations is significant. This project aims to explore the application of speech recognition in lie detection, focusing on improving the accuracy and reliability of the system in a criminal investigation scenario. By analysing a large dataset of speech samples under controlled and naturalistic conditions, the project will identify key vocal indicators of deception and develop a model that can be integrated into secure web page frameworks.

2.2 Project Significance

2.2.1 Current limitations

Current research in lie detection mainly reveals three key limitations in the field of voice-based deception detection: data transparency, black-box model issues, and portability in multimodal systems. These challenges hinder the development and usage of reliable and reproducible systems. Addressing these limitations is critical to advancing the field and ensuring more accurate and ethical applications of lie detection.

2.2.1.a Data Transparency Issues

A significant limitation of numerous existing lie detection projects is the lack of sufficient data transparency. A project conducted by Alice in 2019 experimented an analysis utilising only two sample data points, without providing details regarding the dataset [5]. The lack of transparency in data selection and model training limits the credibility and reproducibility of such projects, making it difficult for other researchers or practitioners to verify the results or apply the same approach to different datasets. This undermines the potential for real-world applications and raises concerns about the reliability of such systems.

2.2.1.b Black-Box Model Problem

A further significant issue is the reliance on black-box models in the field of lie detection. Typically for pre-trained models, they are incorporated into comprehensive applications without sufficient transparency regarding their operational mechanisms or decision-making processes. In the 2021 study by Huang, a pre-trained model was integrated into a graphical user interface (GUI)[6]. However, its absence of comprehensive evaluation, performance metrics, and explainability rendered it challenging for users to comprehend the rationale behind the system's conclusions. The "black-box" phenomenon presents considerable obstacles in high-stakes contexts, where users must have confidence and insight into the system's underlying logic.

2.2.1.c Scalability Issues in Multimodal Systems

The use of multimodal lie detection systems, which integrate voice, text, and facial features, has demonstrated efficacy and potential in the capture of more comprehensive signals of deception. However, the lack of scalability and comprehensive development support represents a significant challenge in terms of real-world applicability. In a 2022 study by Yang, the project combined voice, text, and facial features into a single model but did not demonstrate how the system could be adapted to other environments or datasets[7]. The lack of a robust framework for transferring the model to different use cases represents a significant limitation in the practical application of multimodal approaches.

2.2.2 Proposed Solution and Innovations

To address the above challenges, this project focuses on creating a more transparent, explainable, and adaptable voice-based lie detection system. By using advanced machine learning and deep learning models and integrating them into a user-friendly web application, the project seeks to overcome the limitations of current lie detection projects.

2.2.2.a Model Selection and Soft Voting System

The project introduces a combination of four different machine learning and deep learning models—such as DNN, KNN, and SVM, to analyze voice features extracted from large datasets. Each model processes the data independently and produces a prediction. Also to enhance reliability, a soft voting ensemble system is implemented which is particularly effective in this context as it allows each model to contribute based on its strengths, reducing the likelihood of error from any single model and improving overall accuracy. This system aggregates the prediction probabilities from each model and calculates a weighted average, thereby generating a final classification decision.

2.2.2.b Frontend and Backend System Integration

In addition to model development, this project places a premium on the creation of an intuitive web application that facilitates connectivity between the backend models and a user-friendly frontend interface. This integration will facilitate the straightforward uploading of audio data by users, and the system will automatically analysing the samples in the backend and generate interpretable results. By providing a transparent and accessible web interface, the system enhances the user experience while also ensures that the intricate backend operations remain transparent and manageable.

2.2.2.c Scalability and Explainability

Another innovation of this project is its focus on developing a transferable and explainable model. The aim is to guarantee that the model is capable of functioning with the existing dataset and can also be readily adapted for further updates and new cases. By optimising and simplifing the training pipelines, the model is designed to accommodate audio data from disparate contexts, thereby enhancing its generalisability. The comprehensive analysis of the project pipeline, including data processing, model training and internal decision-making processes, ensures that users are able to comprehend the rationale behind specific predictions. This feature will facilitate the establishment of trust in the system, particularly in critical applications such as criminal investigations.

2.3 Project Content

2.3.1 Project Scope

This project encompasses both academic and applied research, with the objective of developing a voice recognition system that is specifically tailored for the detection of deception in criminal investigation scenarios. The project employs a wide range of intelligent reasoning systems and cutting-edge machine learning techniques to analyse speech patterns and vocal stress indicators, with the objective of accurately distinguishing truth from deception.

This project explores pivotal domains of artificial intelligence (AI), with a concentrated emphasis on machine learning algorithms and deep learning methodologies. The system employs patterns derived from a vast repository of voice data to generate hypotheses and predictions pertaining to deception. By analysing hundreds of voice samples, the system is trained to identify common traits of deceptive speech, thereby enhancing its capacity to predict deception. In addition to inductive reasoning, the project makes use of deductive and abductive reasoning. Deductive reasoning will be employed to evaluate specific vocal indicators that have been identified in existing literature as markers of stress or deception. These include, changes in voice pitch and increased speech hesitations. The system assesses the alignment of a new audio sample with patterns of deception by applying known principles and vocal markers. As for abductive reasoning, it serves to enhance the model by inferring the most probable truth state based on incomplete or partial information. This is a crucial aspect in real-world scenarios where evidence is often fragmented.

2.3.1 Project Challenges

The project is confronted with a number of academic and practical challenges. From an academic perspective, one of the most significant challenges is the accurate collection of data. The project needs to ensure that the sample of audio data used in criminal investigation scenarios is diverse and representative in order to ensure the generalisability of the model. It is imperative that the dataset encompasses a comprehensive range of vocal patterns, dialects, accents and emotional states in order to guarantee that the system is not biased.

From a market and industrial perspective, the scalability of the technology and its acceptance within legal frameworks also present challenges. As the technology is developed for usage in sensitive environments including criminal investigations and courtrooms, it must comply with the relevant privacy legislation and address any potential ethical concerns. Furthermore, the potential for cultural bias in the training of algorithms represents a significant constraint. The system's scalability and cross-domain effectiveness shoule also be given due consideration, as variations in vocal expression across cultures need to be taken into account to prevent inaccuracies or biased conclusions.

To address these challenges, the project will develop robust validation frameworks to test the model’s accuracy and adaptability in real-world conditions. The project will also integrate secure web page frameworks to make the technology accessible to experts in legal, law enforcement, and security settings. This seamless integration enables practical deployment of the technology.

2.4 Business Plan

2.4.1 Business Value

The incorporation of sophisticated, intelligible, and transparent voice recognition technology into the field of lie detection has considerable business value across a range of sectors. As the feature of lying in auodio remains steady in different scenarios, this project provides a initial trial and reference to the market. In the criminal justice system, where the reliability of evidence is of paramount importance, this system can serve as a valuable tool in interrogations and investigations. It provides law enforcement with a non-intrusive, scientifically grounded method for assessing the veracity of statements. Such technology could also prove beneficial to the healthcare sector, particularly in psychiatric evaluations where patient honesty is of the utmost importance. To this end, the systemis designed to possess adaptability and scalability, which render it an attractive option for corporate security and fraud detection, as companies increasingly seek reliable methods for verifying truthfulness in high-stakes situations such as legal disputes or internal investigations. While also by addressing the current deficiencies in data transparency, model explainability, and scalability, this project has the potential to markedly advance the field of lie detection, offering a reliable and practical tool for real-world applications.

2.4.2 Company Vision and Mission

Our company's vision is to become the global leader in the provision of lie detection solutions based on speech recognition technology, supporting industries in enhancing transparency and integrity.

The mission is to assist law enforcement, healthcare organisations and businesses in identifying authenticity in a non-invasive manner, and to improve security and efficiency by providing scientific and interpretable speech recognition tools.

2.4.3 Products and Services

2.4.3.a Core technologies

Speech recognition technology: Using advanced speech recognition algorithms and natural language processing technology, the system is able to extract important information from speech signals.

Lie Detection Algorithm: The system determines the truthfulness of a statement by comparatively analysing data such as the speaker's voice characteristics, intonation changes, speech rate and stress indicators.

Model interpretability and transparency: Our system uses an interpretable model to ensure that users can clearly understand the reasoning behind each judgement. Data transparency helps to increase the credibility of the system and reduce legal risks.

2.4.3.b Product Features

Real-time speech analysis: The system is able to analyse recorded or live speech and make authenticity judgments within seconds.

Contextual adaptation: It provides customised algorithm models for different industries (e.g. legal, medical, corporate) to meet the specific needs of different industries.

Multilingual support: The system can support speech analysis in multiple languages, expanding its application potential in the global market.

2.4.4 Market Analysis

2.4.4.a Market size

According to research, the global speech recognition market is expected to grow at a compound annual growth rate (CAGR) of 17.2% over the next five years. Increasing market demand from criminal justice systems, healthcare organisations and corporate internal audit has created a great need for speech-based lie detection technology.

2.4.4.b Competitive Analysis

Several lie detection technologies based on physiological signals (e.g. polygraph) or behavioural analysis (e.g. facial expression recognition) currently exist in the market, but all suffer from the problems of insufficient accuracy, high intrusiveness and poor interpretability. This system will be a market leader with the advantages of non-invasiveness, high accuracy and high interpretability.

2.4.5 Business Model

2.4.5.a.Revenue streams

- Software-as-a-Service (SaaS): Providing a cloud-based platform for lie detection services to law enforcement agencies, hospitals and corporations on a pay-per-use or monthly subscription basis.
- Customised solutions: Provide personalised and customised services to large corporations and organisations, developing proprietary models and tools for their specific needs and charging development and maintenance fees.
- Partner Programme: Work with organisations in the legal, medical or corporate security fields to jointly promote the product and share the profits.

2.4.5.b.Pricing strategy

First for standard edition, we conducted pay per use, suitable for small and medium enterprises or temporary projects. And for enterprise Edition Subscription-based service, suitable for large organisations for ongoing use.Last for custom service, we provides bespoke functionality for customers with specific needs and charges low service fees.

2.5 Project Objective

2.5.1 Base Line Setting

The primary objective of this project is to develop a highly accurate and scalable voice recognition system for lie detection, capable of addressing the limitations of current technologies while offering a solution that is both adaptable and reliable in real-world scenarios. Drawing upon the findings of DePaulo et al., which reveal that untrained individuals typically achieve a lie detection accuracy of only 54%, with professionals seeing only modest improvements, this project seeks to significantly raise detection accuracy to over 60% [8].

To achieve this, the project utilizes advanced AI technologies and big data analytics. The system is designed to continuously learn and improve through exposure to a wide range of speech data simply by backend maintanance and updating, thus enhancing its ability to detect subtle vocal indicators of deception. A key objective is that the system implement a robust soft voting system that combines the strengths of multiple machine learning models (e.g., Random forest, DNN, KNN, and SVM) to ensure more accurate and reliable results. By employing this method, the system will take advantage of each model's individual strengths, aggregating their predictions to create a more balanced and reliable output as well as minimize the impact of any one model's weaknesses, significantly improving the overall detection accuracy.

2.5.1 Web Application Setting

The project focuses on developing a scalable, user-friendly web application that seamlessly integrates with the backend machine learning models. This application allows professionals, such as law enforcement officers and legal experts, to upload audio samples and receive a detailed analysis in real-time. The system will generate clear, interpretable results, highlighting the vocal features and stress indicators that led to the final decision, ensuring that users can understand the system's decision-making process, thereby fostering trust and ensuring the system's adoption in legal settings.

The project also aims to enhance the scalability of the lie detection system, ensuring that it can be effectively used across multiple domains and industries. By focusing on model explainability and creating transparent workflows for model development and testing, the system will be applicable not only in criminal investigations but also in corporate security, fraud detection, and other sectors where truth verification is critical.

Overall, the project's objectives include:

- Achieving an accuracy rate of over 60% in lie detection using voice recognition.
- Developing a scalable and adaptable system capable of handling diverse voice data from multiple domains.
- Implementing a soft voting ensemble model to improve prediction reliability and reduce error rates.

- Creating a user-friendly web application that offers real-time, interpretable results for professionals.
- Ensuring the system’s compliance with ethical standards, privacy laws, and minimizing cultural biases to facilitate broad adoption.

3. Literature Review

3.1 Relevant Research

- According to Xi et al.(2024), the semi-supervised lie detection algorithm integrates multiple speech emotional features to improve detection accuracy. The model uses Long Short-Term Memory (LSTM) and Auto Encoder (AE) networks for feature extraction and applies a joint attention model for feature fusion. Local Maximum Mean Discrepancy (LMMD) and Jefferys multi-loss optimization further enhance the classification performance. The proposed method improves accuracy by approximately 2-3% compared to traditional methods, with a maximum accuracy of 65.74%[9].
- In Alice Xue's (2019) research, an automated lie detection system was built using ML models trained on acoustic features. The study focused on using Mel-frequency cepstral coefficients (MFCC), energy envelopes, and pitch contours from speech recordings in a two-person lying game. A majority-voting ensemble model combining Gradient Boosting Classifier (GBC), SVM, and SGD achieved a maximum accuracy of 55.8%, outperforming the baseline of 50% and human accuracy of 48%[3].
- Gideon Mendels et al.(2017) presented a hybrid acoustic-lexical deep learning approach for deception detection using the Columbia X-Cultural Deception Corpus (CXD). They compared different ML models with spectral, acoustic-prosodic, and lexical features. A novel deep neural network combining acoustic and lexical streams achieved a 15% improvement in F1-score over a random baseline[10].
- The work by Bareeda et al.(2021) focused on a non-invasive lie detection method utilizing speech processing techniques. The approach involves extracting meaningful features from speech, specifically MFCC, and using an SVM classifier to differentiate between truth and lies. The study used a dataset of real-life trial recordings with 61 deceptive and 60 truthful clips. The system

achieved a classification accuracy of 81% using a polynomial kernel for SVM[11].

- In the research by Al-Dhaher et al. (2024), they developed an approach to lie detection using voice stress analysis. The study explores the extraction of both time and frequency domain features from audio signals, such as MFCC, pitch, and spectral entropy. By using random forest for feature selection and classification, the model was trained and tested on a real-world dataset. Their system achieved an accuracy of 79% in distinguishing truthful from deceptive speech. The research also highlights MFCC as the most significant feature for deception detection[12].
- Fernandes and Ullah (2021) applied ML to detect deception from speech signals using spectral and cepstral features. They extracted features like MFCC, delta cepstrum, and spectral energy to differentiate between truthful and deceptive speech. The study used LSTM and Levenberg-Marquardt algorithms to classify the features. With PCA applied, the time-difference spectral energy feature achieved an accuracy of 100%, while the delta cepstrum feature showed a 91.66% accuracy[13].

3.2 Key Findings

- Emotional, acoustic, and linguistic speech features are crucial to enhance deception detection accuracy.
- Acoustic features such as MFCC, pitch, and spectral energy are significant for distinguishing truth from deception.
- Machine learning models like LSTM, SVM, Random Forest, and hybrid deep learning approaches are commonly used in deception detection systems.
- Performance improvements over traditional methods typically range from 2-3%, with maximum detection accuracies varying from 55% to 81%.
- Ensemble and deep learning approaches show significant improvements in deception detection tasks across various datasets and contexts.

3.3 Methodologies

- Long Short-Term Memory (LSTM) networks: Used for feature extraction in speech-based deception detection.
- Auto Encoder (AE) networks: Used for feature extraction and fusion in semi-supervised models.
- Support Vector Machine (SVM): Used for classifying speech features, especially MFCC, to detect deception.
- Random Forest algorithm: Used for feature selection and classification, particularly with voice stress analysis.
- Majority-voting ensemble models: Combining classifiers like GBC, SVM, and SGD for improved performance.

4. System Design

4.1 Architecture Overview

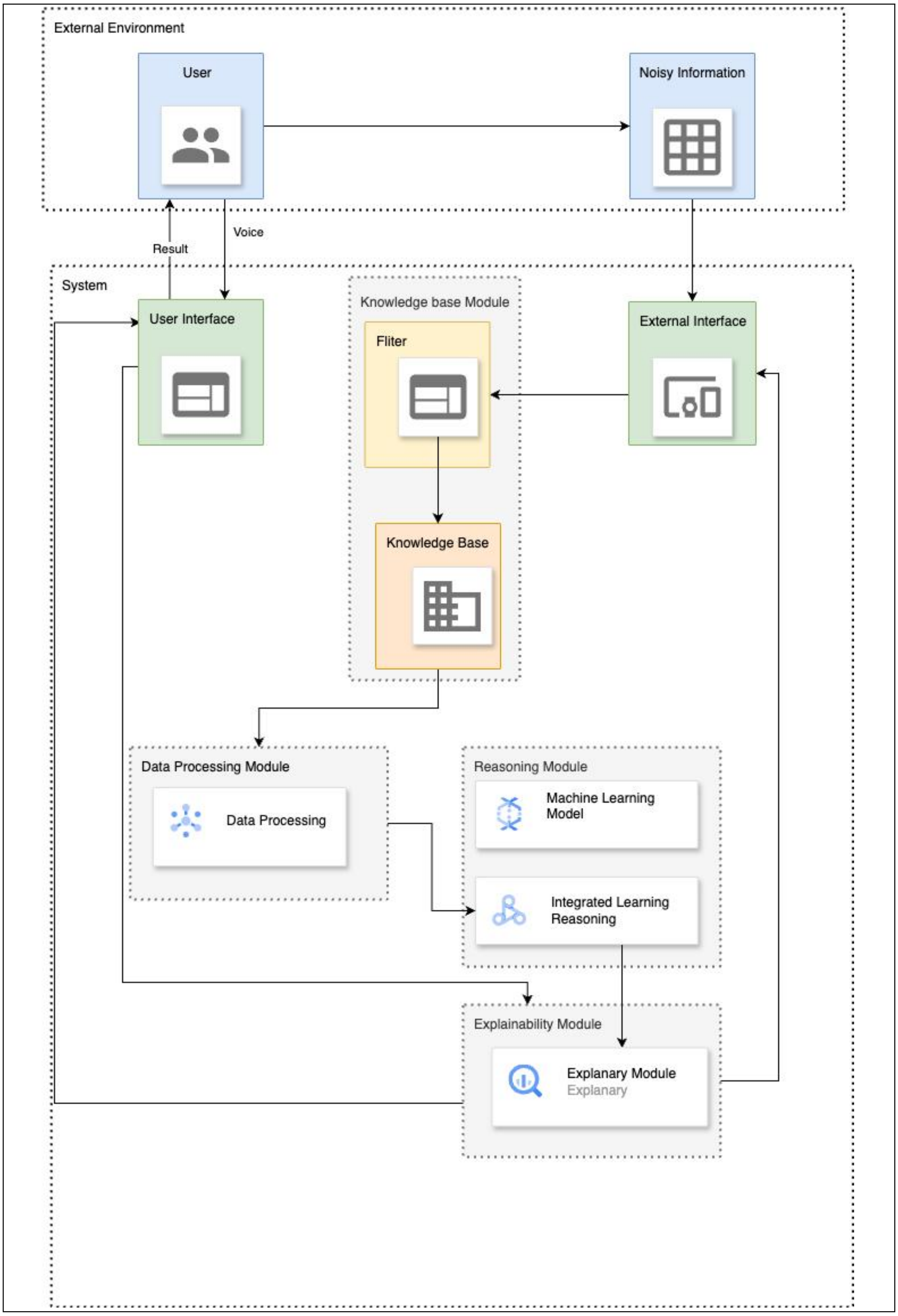


Diagram 4-1: Architecture and flowchart of the project system

The system architecture comprises six principal components: the user interface , the data processing module, the knowledge base module, the reasoning module, the explainability module, and the external interface.

These components serve as the foundation for the internal system. Furthermore, as an exemplary reasoning system, interaction with the external environment is imperative. It is essential that the content of the knowledge base be updated on a continuous basis over time. The system's input is a voice file. (Currently only supported WAV files) The probability of the authenticity of this voice output can be calculated based on the data obtained by the knowledge base through the external interface and the data processing layer. As the data continues to increase, the probability of the voice output being authentic can be calculated with greater accuracy. The structural data and new data obtained by the knowledge base update our reasoning model, which in turn improves the accuracy and credibility of the model and the system's generalisation capabilities. Subsequently, the role of the various levels will be introduced in preliminary form, with a more detailed explanation provided in the subsequent section.The following points will be discussed respectively in next section:

- User interface Module
- Data processing Module
- Knowledge base Module
- Reasoning engine Module
- Explainability Module
- External interfaces

4.2 System Components

In this section we will initially introduce the parts of our system:

4.2.1 User Interface Module

This module aims to provide a user-friendly interface for submitting input and displaying outcomes. We chose the implementation of a web application because it not only facilitates an efficient design and development process but also inherently supports multi-platform devices. The web app is developed using the popular Vue.js framework, which is famous for its flexibility, reactivity, and ease of use. The user interface accepts input by either recording with the microphone or simply uploading an audio file. The files will be converted to WAV format, packed into an HTTP request, and sent to the back-end server for further processing. Once the prediction is generated by the model, the outcome will be sent back to the web app and displayed with a probability attached, indicating whether deception is detected in the input audio.

4.2.2 Data Processing Module

The Data Processing Module contains the retrieval, extraction, conversion, and preparation of the data, ensuring the system can work with clean and well-structured inputs. The process first downloads and extracts specific video segments from the DOLOS dataset, which contains YouTube links and corresponding timestamps for each video. By using YoutubeDL library, it downloads the video content in the required resolution, while FFmpeg extracts relevant segments based on predefined start and end times. It ensures that only the necessary portions of each video are stored for further use, saving them as MP4 files. Once the video segments are prepared, the next step is to convert these video files into audio. The MoviePy library is employed to streamline the process of extracting audio tracks from the video clips. The audio streams are isolated and saved in WAV format, ready for further analysis in the monomodal component of the project. At the same time, the system tracks and logs errors throughout the process, ensuring that missing or problematic files are logged and monitored.

The Data Processing Module is designed to manage large-scale datasets efficiently, with parallel processing capabilities implemented to handle high data volumes. The system processes multiple video files simultaneously, significantly reducing processing time and ensuring that the data is ready for use in the later stages of the project. The final output consists of MP4 video files and their corresponding WAV audio files, forming the foundational data for the project's lie detection models.

4.2.3 Knowledge base Module

This section aims to achieve effective storage, management, and retrieval of information, in order to provide comprehensive support and guidance for users.

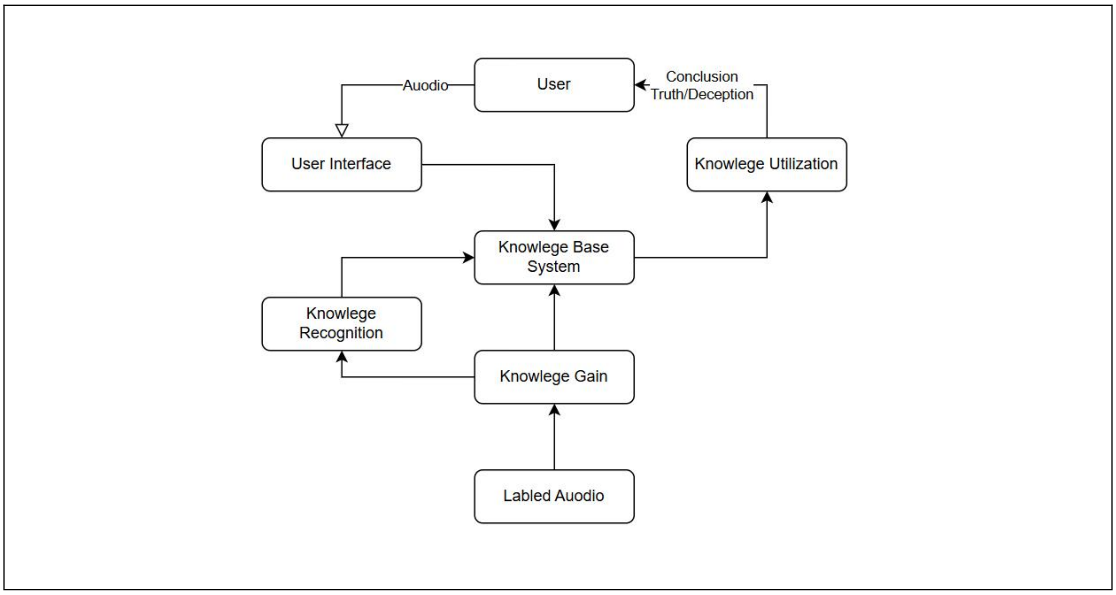


Diagram 4-2: Flowchart of the knowledge base module

As depicted in the Diagram 4-2, the system is designed to manage and process labeled audio data efficiently, converting raw audio data inputs into usable knowledge that can be recognized and utilized. Labeled audio will be fed into the system for knowledge gain, where the audio is analyzed to extract meaningful patterns and insights. This Knowledge Base System enables users to interact with audio data in a structured manner, ensuring the conclusions drawn— truth or deception—are based on well-managed information. The ability to handle data consistently, through both the recognition and utilization phases, makes it an essential component for efficient retrieval and user support.

By using Django's modeling system, the knowledge base can store user uploaded audio files (such as WAV files), ensuring centralized management of all important data and facilitating subsequent access. The knowledge base allows users to easily upload and manage audio files. Each file comes with an upload time for easy tracking and organization of content. In addition, users can quickly retrieve the required audio materials, improving the efficiency of information acquisition. With the powerful query function of Django, users can easily search for files based on different criteria.

4.2.4 Reasoning engine Module

This section is concerned with the process of training our inference engine. Once well-specified data has been obtained from the knowledge base, the data is subjected to secondary processing in order to extract the requisite feature information. Subsequently, general machine learning and deep learning algorithms are selected and fine-tuned in order to optimise their suitability for the task at hand. Following the training and testing of the models to ensure that they neither underfit nor overfit, the models will be placed in the repository. Finally, the learning of these models will be integrated to create a hybrid inference system that combines the strengths of each model. This hybrid inference system will then be connected to the next system module (Explainability Layer).

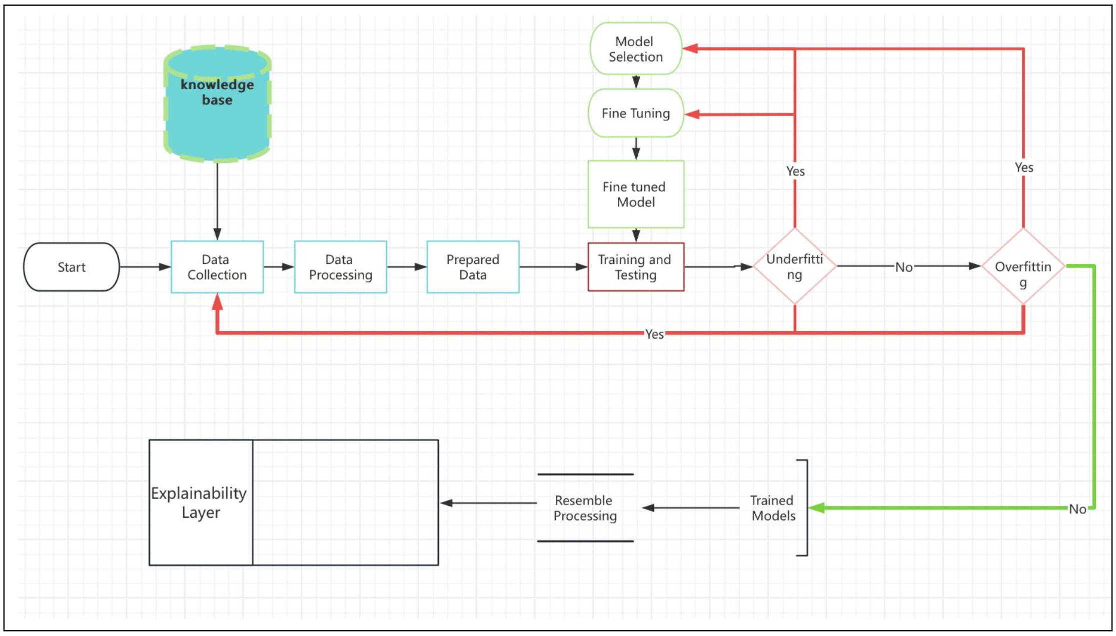


Diagram 4-3: Training process of the reasoning engine

4.2.5 Explainability Module

The interpretation module is comprised of two principal components. The initial component entails the acquisition and assessment of a trained and evaluated model. The subsequent component entails the generation of results through the aforementioned model.

This module receives the trained and evaluated model from the Reasoning Module, which should be the best evaluated model. In this instance, an integrated model is accepted; a detailed description of this will be provided subsequently.

After the backend receives the incoming data from the frontend, the view function in the Django framework processes the data and calls the previously trained model for prediction, and then outputs the results to the interface.

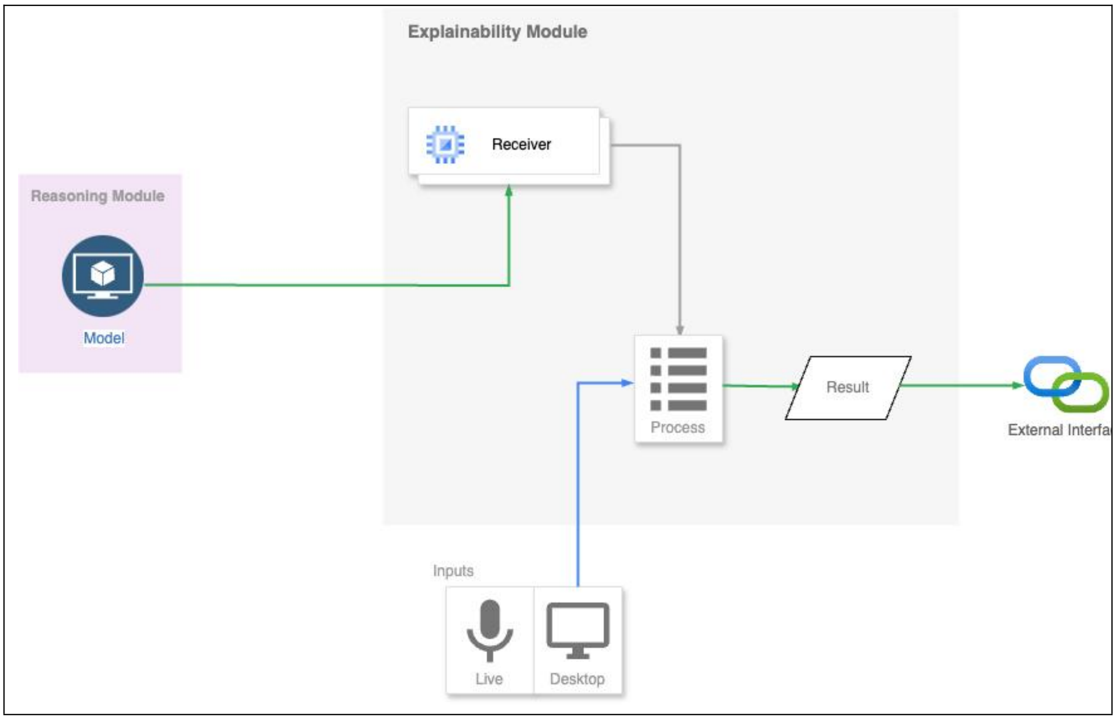


Diagram 4-4: The architecture of Explainability Module

4.2.6 External interface Module

The backend of this project, a RESTful interface has been defined for handling audio file uploads. This interface uses the POST method, allowing users to upload WAV audio files in multipart form data format. After successful upload, the interface will return the status of 201 Created and a success message. If an error occurs, detailed error information will be returned for the user to correct. The logic within the interface also provides scalability for future audio processing, allowing developers to conduct further audio analysis or feature extraction after file upload.

4.3 Reasoning Techniques and Algorithms

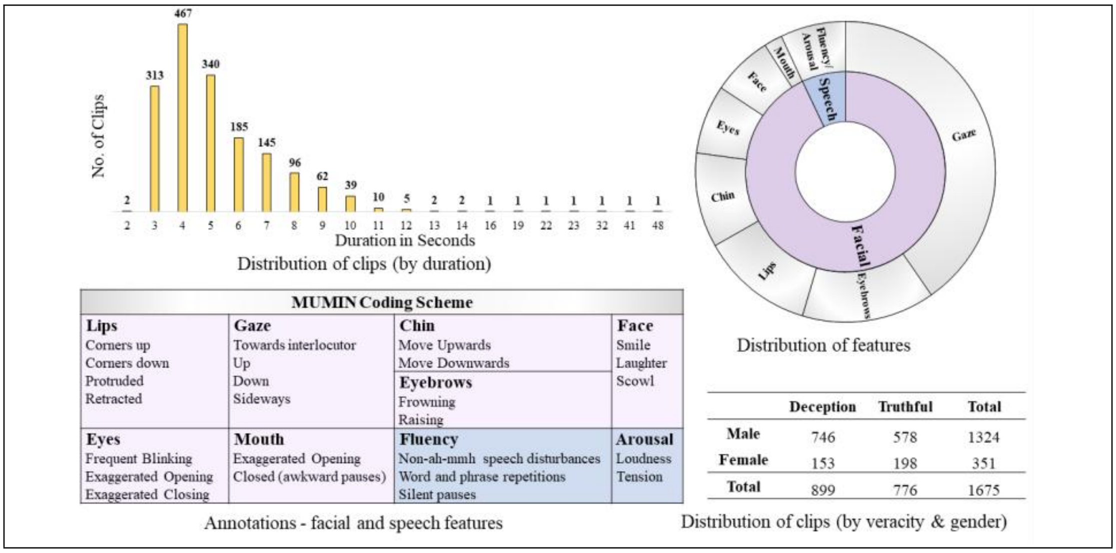
The inference engine of our system is trained using a machine learning model based on well-labelled data from the knowledge base. This is followed by inductive inference to test the accuracy of the inference model. Our inference technique is an integrated hybrid technique that includes Random Forest, SVM, DNN and other deep learning techniques and machine learning techniques. In the subsequent sections, we describe the deployment and integration of these techniques in detail.

5. Data Collection and Preparation

5.1 Data Sources

This project utilizes the DOLOS dataset collected by ROSE lab, a comprehensive and multimodal dataset designed for deception detection research [14]. The dataset is sourced from the reality-TV gameshow "Would I Lie To You?", which provides an authentic setting for capturing deceptive behaviors in participants. The gameshow focuses on naturalistic, unscripted speech in a competitive, high-pressure environment, therefore it closely mirrors the types of deception encountered in real-world investigative and legal scenarios. This makes the DOLOS dataset particularly well-suited for training machine learning models aimed at detecting deception scenarios like law enforcement and courtroom settings.

The dataset contains rich multimodal data, including both visual and vocal features, annotated for research in deception detection.



MUMIN Coding Scheme			
Lips Corners up Corners down Protruded Retracted	**Gaze** Towards interlocutor Up Down Sideways	**Chin** Move Upwards Move Downwards **Eyebrows** Frowning Raising	**Face** Smile Laughter Scowl
Eyes Frequent Blinking Exaggerated Opening Exaggerated Closing	**Mouth** Exaggerated Opening Closed (awkward pauses)	**Fluency** Non-ah-mmh speech disturbances Word and phrase repetitions Silent pauses	**Arousal** Loudness Tension

Annotations - facial and speech features

	Deception	Truthful	Total
Male	746	578	1324
Female	153	198	351
Total	899	776	1675

Distribution of clips (by veracity & gender)

Figure 5-1: Data Overview [15]

Figure 5-1 shows the overview of dataset structure. The dataset includes a total of 1675 video clips, which are extracted from 84 episodes of the reality-TV gameshow. The participants consist of 141 males and 72 females, ensuring diversity in both vocal and facial characteristics. And the video clips vary in length, ranging from 2 to 19 seconds, with the majority having a median duration of 5 seconds. This diversity in duration and participant demographics ensures a comprehensive dataset that can support detailed research in deception detection.

The DOLOS dataset is manually annotated using the MUMIN coding scheme, covering both facial features and vocal features.

Facial features include directional gaze cues (looking up, down, or toward the interlocutor), lip movements (turning up, down, retracted, or protruded), and eye behaviors (blinking frequency, exaggerated openings or closings). Additional facial cues include mouth openness, chin movements, eyebrow raises or frowns, and general facial expressions like smiling or scowling, which are often linked to emotional states or deception. The dataset also captures body gestures, including head movements, hand gestures, arm motions, and shoulder shifts, which play a key role in analyzing non-verbal cues of discomfort or deceit.

Vocal features include fluency issues such as speech disturbances (e.g., non-ah-nmm sounds, repetitions, silent pauses) and arousal markers like loudness and vocal tension, which are indicators of stress linked to deception.

5.2 Challenges in Data Collection

The DOLOS dataset's transparency and feature diversity, with detailed annotations for both facial and vocal features and accessible source material, making it ideal for developing multimodal deception detection models. However，the effective uasage of dataset mainly encounters three challenges regarding the collection and preparation of the data.

The original dataset provided by ROSE Lab includes only video URLs, timestamps, and a corresponding multimodal feature Excel sheet. While this information is crucial for analysis, the dataset lacks directly accessible media files, requiring team members to manually retrieve the videos from YouTube. This is both time-consuming and prone to errors, especially if the video URLs or the timestamps are outdated or inaccurate. The absence of pre-processed media files necessitates additional steps in the project workflow, increasing the overall workload.

In terms of audio extraction for momomodal analysis, although the DOLOS dataset is designed for multimodal deception analysis, the initial focus of this project is on monomodal analysis, particularly audio-based lie detection. To facilitate this, the team must extract audio data from the video clips based on the provided timestamps. Developing an automated, accurate process for isolating and extracting audio is crucial, as any misalignment between the audio extraction and the timestamps could lead to inaccurate results during model training.

Another challengeis the availability of the videos on YouTube. Since the dataset relies on external video sources, some content may become unavailable over time due to copyright restrictions, content removal by the uploader, or other issues. If essential video clips are no longer accessible, this will impact the completeness of the dataset, potentially affecting both data analysis and model training. If huge clips of videos are missing or their length has changed, the team will need to find alternative data sources or deal with data gaps, which could slow down the project's progress.

5.3 Preprocessing Techniques

As for the preprocessing phase, it mainly contains three critical tasks, including downloading videos, transforming videos to audios, extracting audio features.

5.3.1 Video Data Retrieval Process

To streamline the retrieval of video segments, based on the URLs and timestamps provided in the dataset, the first step in preprocessing involves the retrieval of video data from YouTube. To automate the process of downloading and extracting the relevant segments, we utilized a combination of YoutubeDL and FFmpeg libraries.

Based on the CSV file containing metadata for each video (YouTube ID, file name, start and end times), CSV parsing and timestamp conversion was parsed, and the timestamps provided in “MM“ format were converted into seconds to ensure compatibility with FFmpeg for video processing. The YoutubeDL library was used to download each video in the appropriate resolution, and FFmpeg extracted the specified segments based on the start and end times, saving them as individual MP4 files for analysis. To manage the large volume of videos, parallel processing was implemented using the ThreadPool library, allowing multiple videos to be processed simultaneously, which significantly reduced overall processing time. Any videos that were unavailable (e.g., due to copyright issues) were logged for review, and errors encountered during the download or extraction process were tracked to ensure dataset consistency and accuracy. Upon completion, the extracted video segments were saved as MP4 files, which were then used for subsequent audio extraction and analysis in the project.

5.3.2 Video Data Convert Process

After downloading and extracting the necessary video segments, the next step involved converting the video files into audio format for further analysis. Since the focus of the initial research phase is on monomodal analysis—specifically audio-based lie detection—it was essential to extract the audio tracks from the video clips.

The conversion workflow began by loading the pre-downloaded video segments from the designated directory. Using MoviePy, each video file was opened, granting access to the embedded audio stream. The audio stream was extracted using MoviePy's audio processing functionality, isolating the audio component for further analysis. If a video file contained no audio, the system logged a message to track the missing data. After extraction, the audio files were saved in WAV format in the specified output directory, with file names mirroring the corresponding video file names to ensure easy linking between audio and video segments. In cases where no audio was detected, the system logged the absence of audio to help monitor and address any potential gaps in the dataset. Once the video-to-audio conversion was completed, all extracted audio files were saved in WAV format for monomodal analysis of vocal features, a key aspect of the project's lie detection research. The original video files in MP4 format were retained for further feature extraction and analysis in subsequent stages.

5.3.3 Feature Extracted

In this subsection, we explain how to clean the description files and how to perform feature extraction.

After obtaining Nogan's audio slices and their corresponding labels, we will perform feature extraction and processing on these audio files, but before that we need to clean the Anotation file, which are labelled with the labels of each audio clip and other information, but the label data is too cluttered, so we hope that at the end there are only two kinds of labels presented, ***‘truth’*** and ***‘deception’.*** The original tag contains a number of tags such as *‘ truth,’ ‘Truth‘, ‘lie,’ and NA.* For the cleaning part we have the following steps:

- Change all characters to lowercase
- Remove all spaces
- Replace all *‘lie’* and *‘lying’* with ***‘deception’.***
- Deletion of samples labelled NA

After going through the above cleaning steps, we have a clean Anotation file.

MFCC conversion and coding

Mel-frequency cepstral coefficients (MFCC) represent a prominent feature extraction method employed across numerous domains within audio and speech signal processing[16]. By emulating the auditory system of the human ear, MFCC effectively extracts pivotal information from audio signals, particularly the frequency distribution and energy characteristics of the sound. Consequently, MFCC has become a pervasive technique across diverse applications, including speech recognition, sentiment analysis, and speaker identification.

The audio clips were processed in batches using MFCC, with thirteen features or cepstrum coefficients extracted. These represent different aspects of the frequency distribution. The low-order coefficients (e.g. C1, C2) capture the general spectral shape of the audio, reflecting the main features of the speech. In contrast, the higher-order coefficients (e.g. C11, C12) contain more details[17], reflecting more subtle frequency variations.

At last, the tags are aligned with the MFCC feature values of the corresponding audio, and they are encoded in a way that assigns the value of ***1 to truth*** and ***0 to deception***. At this point in the process, the features extracted by the MFCC are normalised, resulting in a data frame. This concludes the data processing phase.

6. Implementation

6.1 Platform and Tools

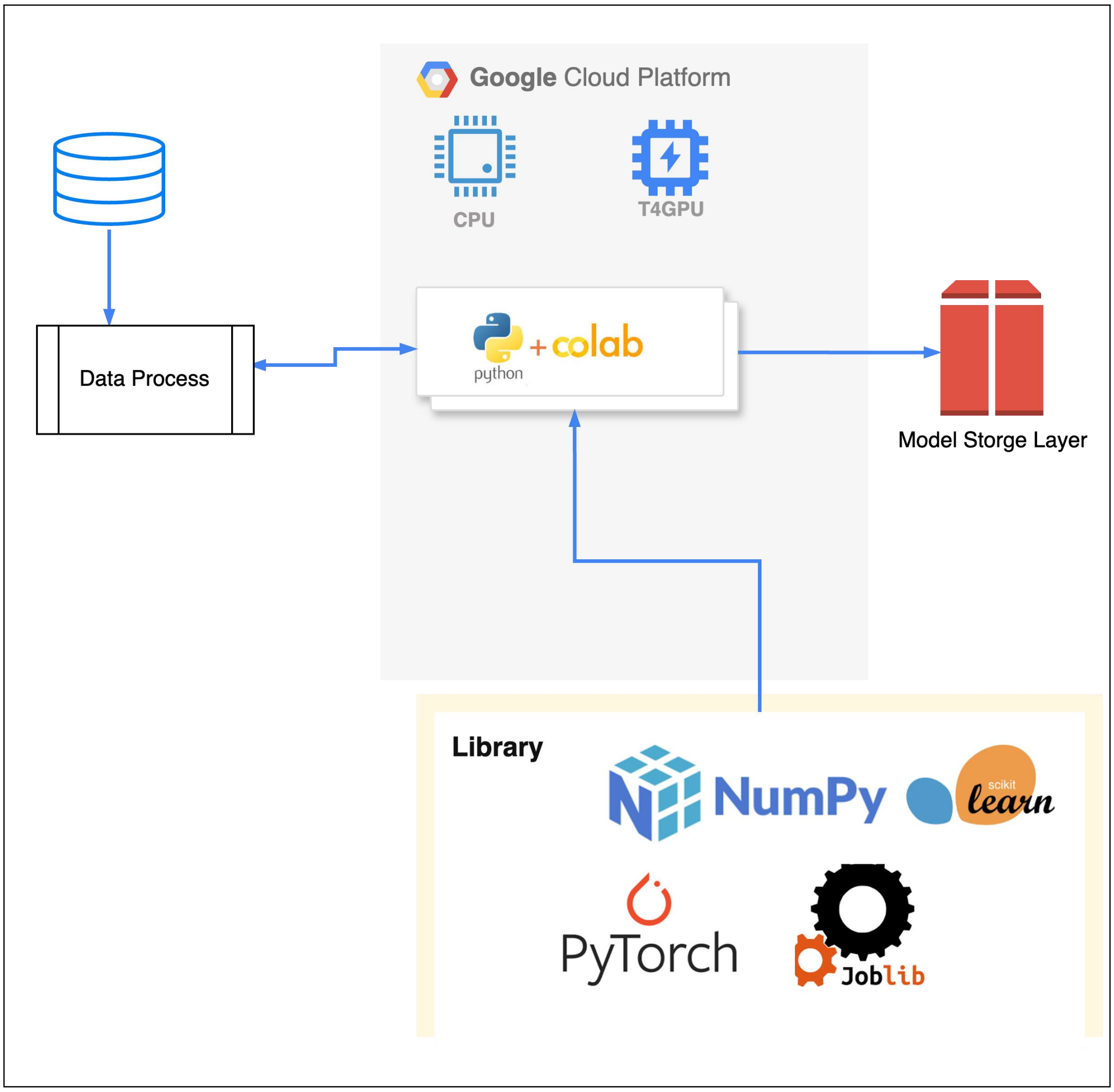


Figure 6-1: Platform and tools for model training

- **Platform**

Regarding model training, all data feature extraction as well as model training is done on colab, by uploading data to Google Drive and then connecting it on colab, using CPU as well as T4 GPU accelerated training.

- **Tools**

We adapted a number of packages and frameworks to be used as tools to help train the models based on python, using pytorch to train the deep learning models, the sklearn package to help tune and train the machine learning models, and Joblib to help store and download the models. numpy was mainly used to standardise some of the data formatting issues. In addition, there are many other packages such as pandas, matplotlib, etc. to help us train and evaluate the models.

6.2 Methods and Technologies

In selecting the technical approach, machine learning and deep learning methods are employed, along with the soft voting technique of integrated learning. This is trained to form the inference part of the whole inference system. Soft Voting is an integrated learning technique, which is mainly The technique is employed for the purpose of integrating the prediction outcomes of multiple classifiers, with the objective of enhancing the overall accuracy and stability of the system. In this system, the soft voting approach facilitates superior robustness and accuracy in the detection of lies, and enhances the reliability of the results [18].

Subsequently, the machine learning methods employed will be delineated individually.

6.2.1 Random Forest

Random Forest method is an integrated learning approach that is primarily utilized for classification and regression tasks. It enhances the precision and resilience of models by integrating the predictions of multiple decision trees [19]. The fundamental concept of Random Forest is to generate multiple decision trees through the two pivotal concepts of "random" and "forest," and then merge their predictions to arrive at a final decision.

As a stable machine learning technique, Random Forest is applied to our training model as a Random Forest Classifier. The data has been processed with MFCC to obtain 13 features,the inputs to the subsequent learning training methods are also all 13 features from this decomposition. This section will also demonstrate the hyper parameter settings of the training model and the results of the test.

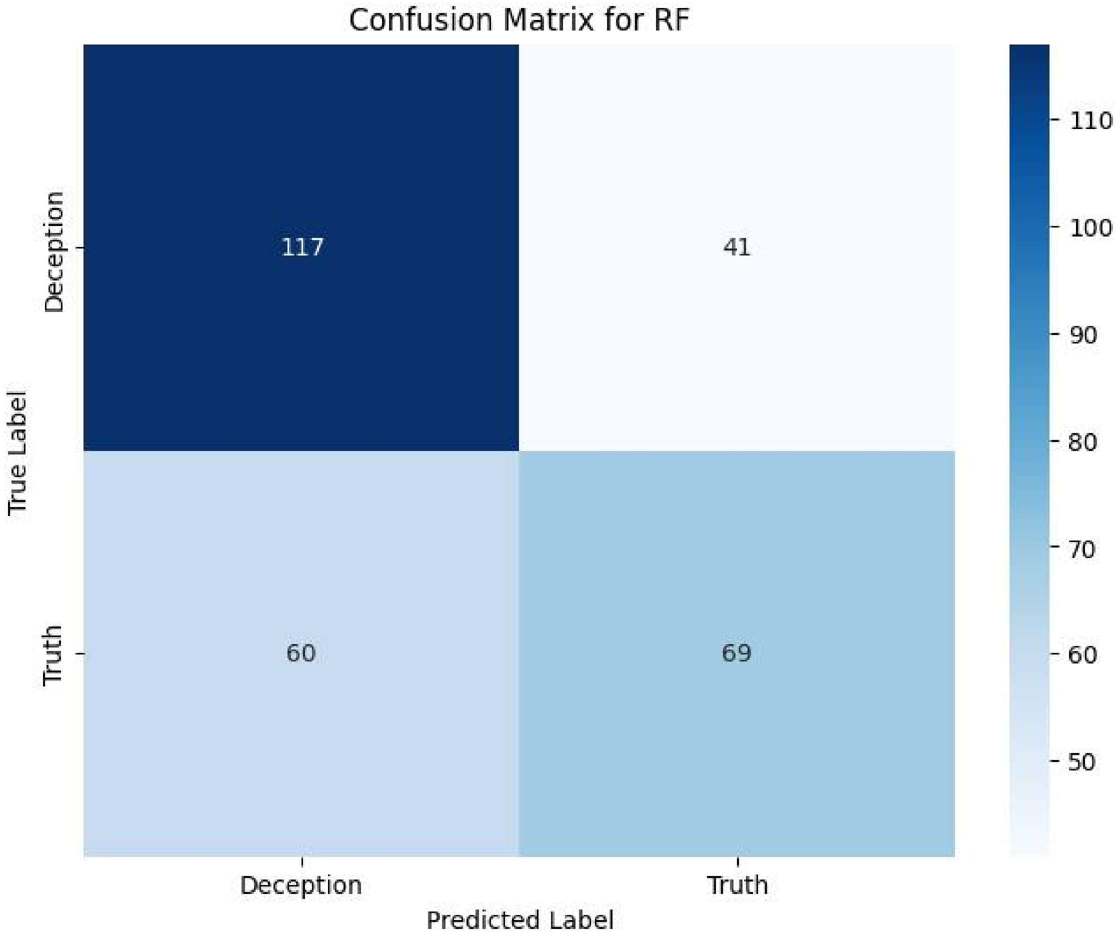


Figure 6-2: Confusion matrix classification plot of the results of the random forest classifier on the test set

A Random Forest Classifier with 1000 trees has been constructed using the Random Forest Classifier function. The value of n_estimators=1000 denotes the number of trees in the forest, while random_state=42 is employed to ensure the repeatability of the results. Following the training phase, the trained Random Forest Classifier was tested on a separate set of data, achieving an accuracy of 62.37%. This result exceeds the baseline accuracy typically observed in our previous publications and is deemed to be a promising outcome. Described for Diagram 1, With regard to the category of deception, the model correctly identified 114 samples, but incorrectly identified 44 samples as belonging to category 1. With regard to the category of truth, the model correctly identified 65 samples, but incorrectly identified 64 samples as belonging to category 0. The model demonstrates superior performance on the deception category (64% precision, 72% recall) and relatively inferior performance on the truth category (60% precision, 50% recall). In other words, this random forest classifier demonstrates a slight propensity for discerning deception.

6.2.2 Support Vector Machine

SVM is a binary classification model whose basic idea is to separate samples of different classes by a hyperplane [20]. This hyperplane acts as a linear classifier, aiming to maximize the distance between the two classes in feature space. For nonlinear problems, SVM employs a kernel function to map input samples into a higher-dimensional space where they can be separated linearly. This characteristic makes SVM widely used due to its high accuracy and robustness.

For audio lie detection involving complex nonlinear features, SVM effectively improves classification capability by mapping input samples to a high-dimensional space and utilizing kernel functions to find an appropriate hyperplane to distinguish between true and false statements. This approach enhances the accuracy of lie detection and allows for better handling of the intricacies of speech signals, making SVM a valuable tool in this domain.

In our hyperparameter tuning process for the SVM model, we optimized the SVM model by tuning three key parameters: C, gamma, and the kernel function. A grid search with 5-fold cross-validation was used to explore different regularization strengths (C), the local influence of support vectors (gamma), and both nonlinear (Radial Basis Function, RBF) and linear kernel functions. After testing different ranges for these parameters, we found the optimal combination and trained the final classifier accordingly.

The accuracy of SVM Classifier on the test set is as follows:

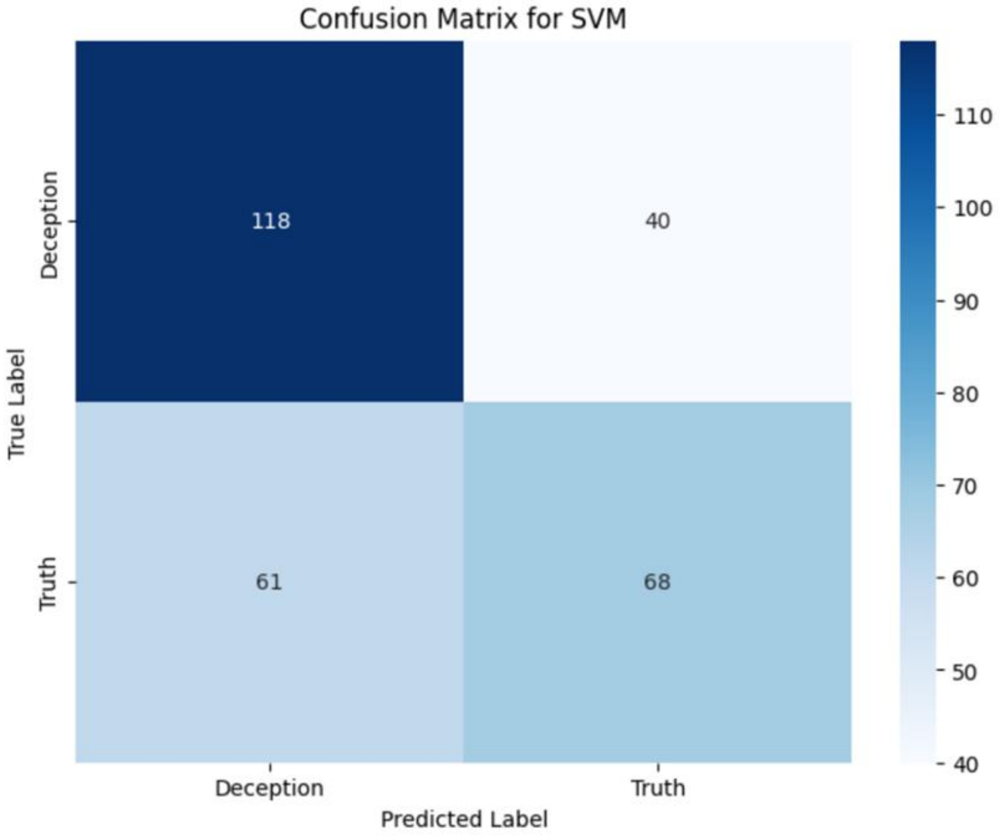


Figure 6-3: Confusion matrix classification plot of the results of the SVM classifier on the test set

The Support Vector Machine performed better overall, with an accuracy of around 65%. It excelled in recall for the deception class (75%), making it adept at identifying deceptive instances, although it had a lower recall for truth (53%). This suggests SVM is a good choice for detecting deceit but may still miss some truthful instances.

6.2.3 K-Nearest Neighbors

The K-Nearest Neighbors (KNN) is an instance-based learning method, or lazy learning method, which does not learn a decision boundary from the training dataset, but directly memorizes the training dataset[21]. In the classification task, KNN selects the K training samples with the closest distance by calculating the distance (e.g., Euclidean distance) between the samples to be classified and the training samples, and then decides the class of the samples to be classified based on the classes of these training samples (e.g., majority voting). KNN is suitable for small datasets and low dimensionality of the feature space.

The choice of n_neighbors has a great impact on the classification results, as it determines how smooth the model is within the localized data region. If n_neighbors is too small, the model may be susceptible to noise; if it is too large, it may ignore fine structure in the data. After experiments and comprehensive consideration, we choose n_neighbors=5, which indicates that the KNN classifier will consider the 5 nearest neighbors when making predictions.

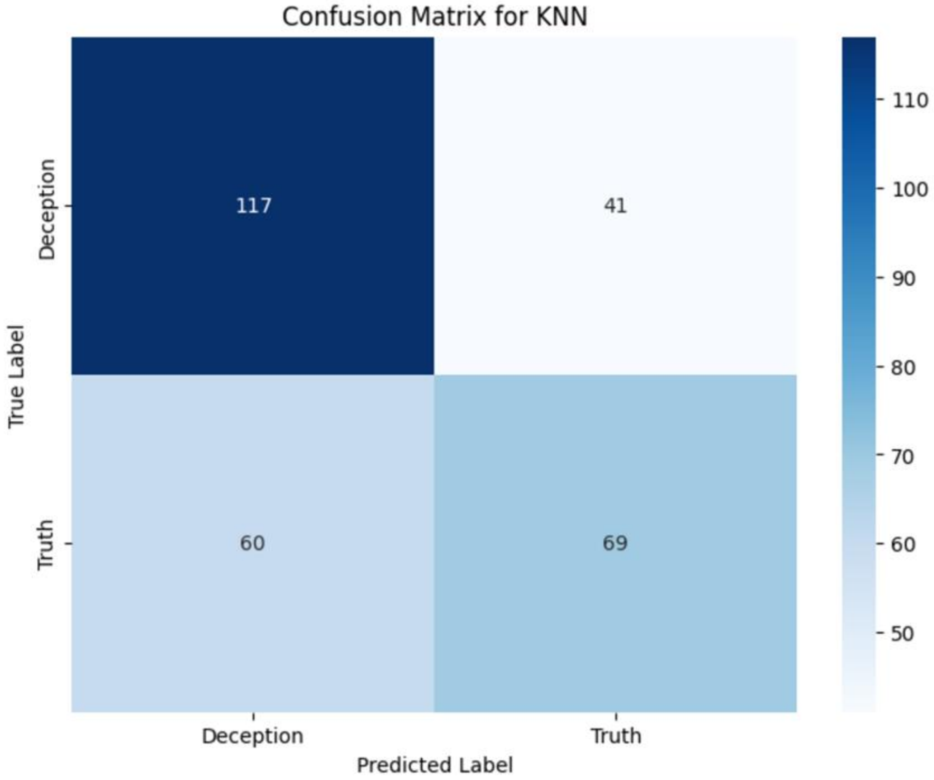


Figure 6-4: Confusion matrix classification plot of the results of the KNN classifier on the test set

The K-Nearest Neighbors algorithm yielded a similar accuracy of about 64.8%. It also showed strong recall for deception (74%) but mirrored the SVM's challenges with the truth, maintaining a recall of only 53%. Thus, while KNN is effective for identifying non-deceptive cases, it shares the same limitations in recognizing truths as SVM.

6.2.4 Deep Neural Networks

The Deep Neural Networks (DNN) is a neural network model that contains multiple hidden layers and is capable of learning high-level feature representations of the input data[22]. DNN optimizes the network parameters through a back propagation algorithm to minimize the loss function of the network on the training set. Its powerful feature learning capability enables it to handle complex nonlinear problems.

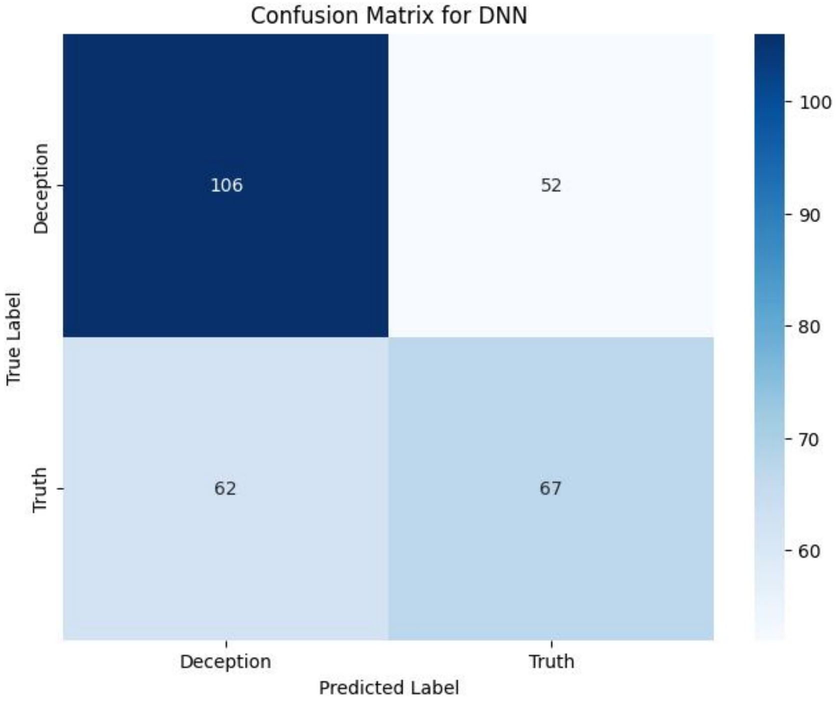


Figure 6-5: Confusion matrix classification plot of the results of the DNN classifier on the test set

The Deep Neural Network was the least effective among the four models, with an accuracy of only 60%. Although it managed a recall of 67% for deception, its performance for truth was notably weaker at 52%. This indicates that while DNN can identify some deceptive cases, it struggles significantly with non-deception detection.

7. Results and Progress

7.1Preliminary Results from Reasoning Engine

In the preceding section, we analysed the impact of each of the distinct machine learning and deep learning models. In this chapter, we evaluate the integrated models to ascertain their effect.

Firstly, the Soft Voting Integration Model with Deep Learning DNN network was added, which is the integration model that includes both Deep Learning and Machine Learning models(RF, KNN, SVM,DNN). This model is referred to as the **Soft Voting Integration Model_A**. In addition, we also test the integrated learning without deep learning DNN network model, that is, we refer to the soft voting integration model which includes three machine learning models (RF, KNN, SVM), we call it **Soft Voting Integration Model_B**. Next, we put both Soft Voting Integration Model_A and Soft Voting Integration Model_B into the test set for testing and evaluation, and analyse the evaluation results.

7.2 Performance Metrics Visualizations

7.2.1 Soft Voting Integration Model_A

Class	Precision	Recall	F1-Score	Support
0(Deception)	0.62	0.86	0.72	158
1(Truth)	0.68	0.36	0.47	129
Accuracy			0.64	287
Macro Avg	0.65	0.61	0.6	287
Weighted Avg	0.65	0.64	0.61	287

Table 7-1: Evaluation performance of Model_A

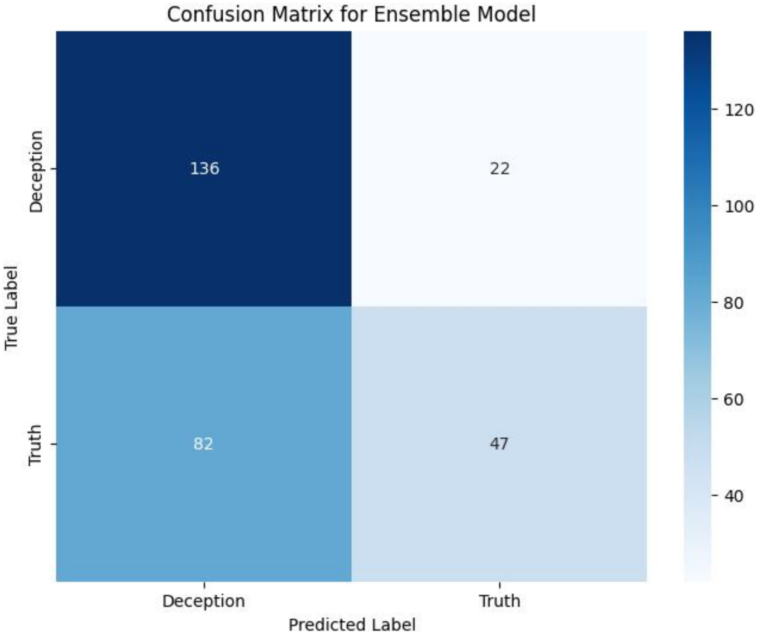


Figure 7-1: Evaluation Confusion Matrix for Soft Voting Integration Model_A

The results of the classification task for the Soft Voting Integration Model_A demonstrate the performance evaluation for both the deception and truth categories. The overall accuracy of the model is 0.6376, which evinces its capacity to make predictions with a reasonable degree of accuracy on the test set.

With regard to category 0 (deception), the model exhibits a precision of 0.62, a recall of 0.86, an F1 score of 0.72, and a support of 158. This indicates that the model is relatively proficient at identifying deception, particularly in terms of recall, which involves identifying a greater number of genuine instances of deception. Conversely, category 1 (real) exhibits a precision of 0.68, a recall of only 0.36, an F1 score of 0.47, and a support of 129. This demonstrates that the model is deficient in recognising genuine cases, particularly in recall, where a considerable number of genuine cases are not correctly identified.

In comparison to the predictive efficacy of preceding individual models, the Soft Voting Integration Model_A does not demonstrate a notable enhancement in prediction accuracy.

7.2.2 Soft Voting Integration Model_B

Class	Precision	Recall	F1-Score	Support
0(Deception)	0.96	0.99	0.98	158
1(Truth)	0.99	0.95	0.97	129
Accuracy			0.98	287
Macro Avg	0.98	0.97	0.98	287
Weighted Avg	0.98	0.98	0.98	287

Table 7-2: Evaluation performance of Model_B

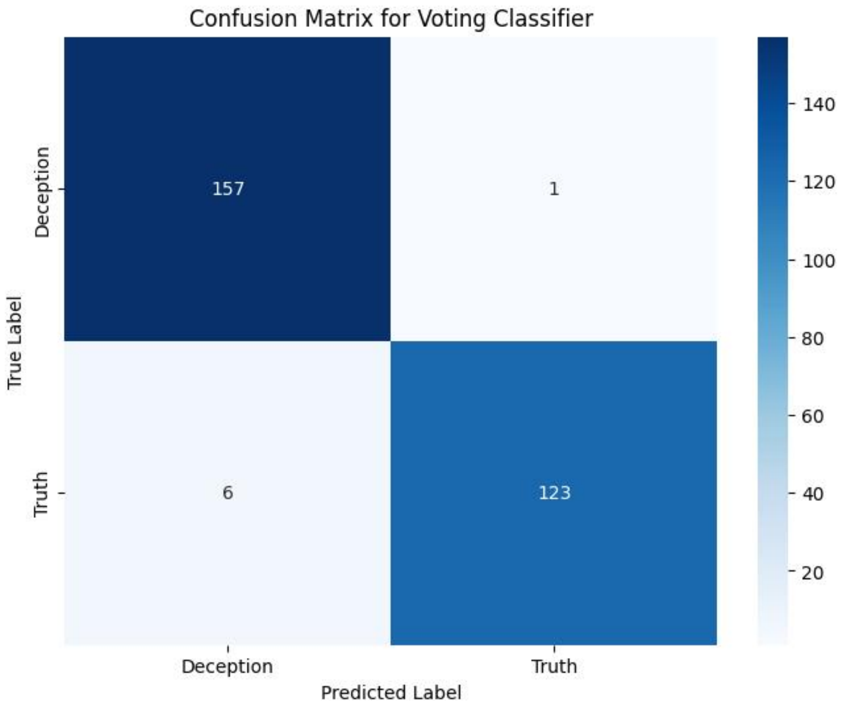


Figure 7-2: Evaluation Confusion Matrix for Soft Voting Integration Model_B

It is noteworthy that the Soft Voting Integration Model_B has demonstrated superior performance compared to previous models, with a notable margin of improvement.

The following figure illustrates the performance of the model in identifying the categories 'deception' (labelled 0) and 'truth' (labelled 1). The overall accuracy is 0.9756, indicating that the model exhibits high predictive power with respect to the test set.

With regard to category 0 (deception), the model exhibits a precision of 0.96, a recall of 0.99, an F1 score of 0.98, and a support of 158. This demonstrates that the model is highly effective in identifying instances of deception and is capable of accurately detecting genuine cases of deception.

For category 1 (truth), the precision is 0.99, the recall is 0.95, the F1 score is 0.97 and the support is 129. This demonstrates that the model also performs well in identifying genuine cases, although with a slightly lower recall rate in comparison to the deception category.

Furthermore, the macro and weighted average metrics demonstrate a macro average precision of 0.97, a recall of 0.98, and an F1 score of 0.97, while the weighted average exhibits a precision, recall, and F1 score of 0.98. These findings substantiate the model's overall efficacy in categorising both categories.

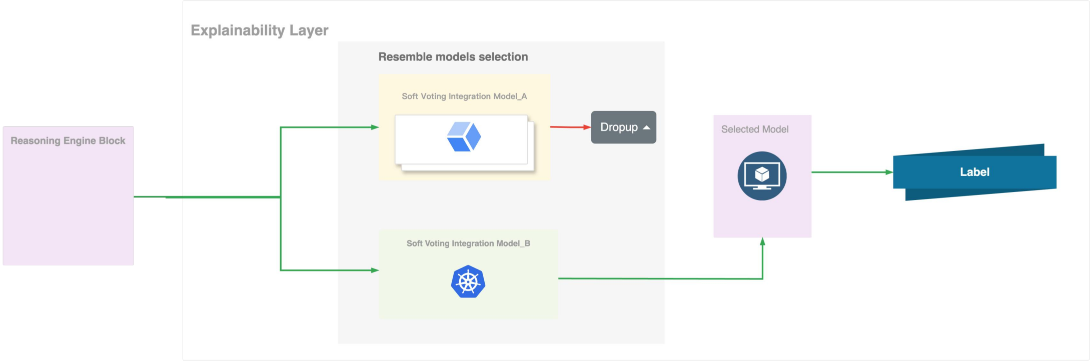

Explainability Layer
Resemble models selection
Soft Voting Integration Model_A
Dropup
Reasoning Engine Block
Selected Model
Label
Soft Voting Integration Model_B

Figure 7-3: The internal structure and flow of the explainability layer

7.3 Comparison with Existing Lie Detection Techniques

In the preceding section, four independent machine learning models and two integrated models were evaluated. This section will focus on analysing and comparing the differences and advantages of these techniques.

Model Name	Label	Precision	Recall Rate	f1-score	Accuracy

RF_Model	Truth	0.60	0.50	0.55	62.37%
	Deciption	0.64	0.72	0.68	
SVM_Model	Truth	0.63	0.53	0.57	64.81%
	Deciption	0.66	0.75	0.70	
KNN_Model	Truth	0.63	0.53	0.58	64.80%
	Deciption	0.66	0.74	0.70	
DNN_Model	Truth	0.56	0.52	0.54	60.28%
	Deciption	0.63	0.67	0.65	
Model_A	Truth	0.68	0.36	0.47	63.76%
	Deciption	0.62	0.86	0.72	
Model_B	Truth	0.99	0.95	0.97	97.57%
	Deciption	0.96	0.99	0.98	

Table 7-3: Performance comparison of models

In terms of both recall and precision, the Soft Voting Integration Model_B is demonstrably superior. This outcome prompted us to conduct additional tests, which were conducted side by side and revealed the same outcome. The integrated learning model thus emerges as the clear winner. However, the Soft_Voting_Integration Model_A did not perform as well. Even in terms of accuracy, it did not reach the same level as the separate knn_model and svm_model. It may be posited that the uncertainty inherent to deep learning models impedes the accuracy of machine learning in general, thereby limiting the potential for significant improvement in accuracy. This issue may be addressed in future research, although it is not the focus of this report.

In terms of accuracy, the KNN_model and SVM_model demonstrate a similar performance, with the RF model exhibiting a comparable outcome. The deep learning model, DNN, however, exhibits a notable decline in performance, which may be attributed to the limited number of parameters and features, particularly when applied to a smaller dataset.

8 Web Application Development

8.1 Initiation

Based on the previous experiments and analyses, the web system was designed and predicted using the previously trained model. This chapter describes the development and implementation process of the system and provides details from the front and back end respectively.

8.2 Front-End Development

The user interface of our deception detection system is an interactive web application built with Vue 3 and Vite. The front-end collects user input, sends it to the back end as an HTTP request, gets a response, and presents it in a clear and readable way.

8.2.1 Tools

Vue.js

Vue is a progressive JavaScript framework for building user interfaces. It is known for its flexibility, reactivity, and ease of use. It builds on top of standard HTML, CSS, and JavaScript and provides a declarative, component-based programming model that helps efficiently develop user interfaces.

Vite

Vite is a modern, fast-build tool designed to improve the development experience, especially for front-end projects. It provides a rapid, lightweight development server with hot module replacement (HMR) and fast production builds. Vite works out of the box with Vue 3 and significantly improves the efficiency of our web development process.

PrimeVue

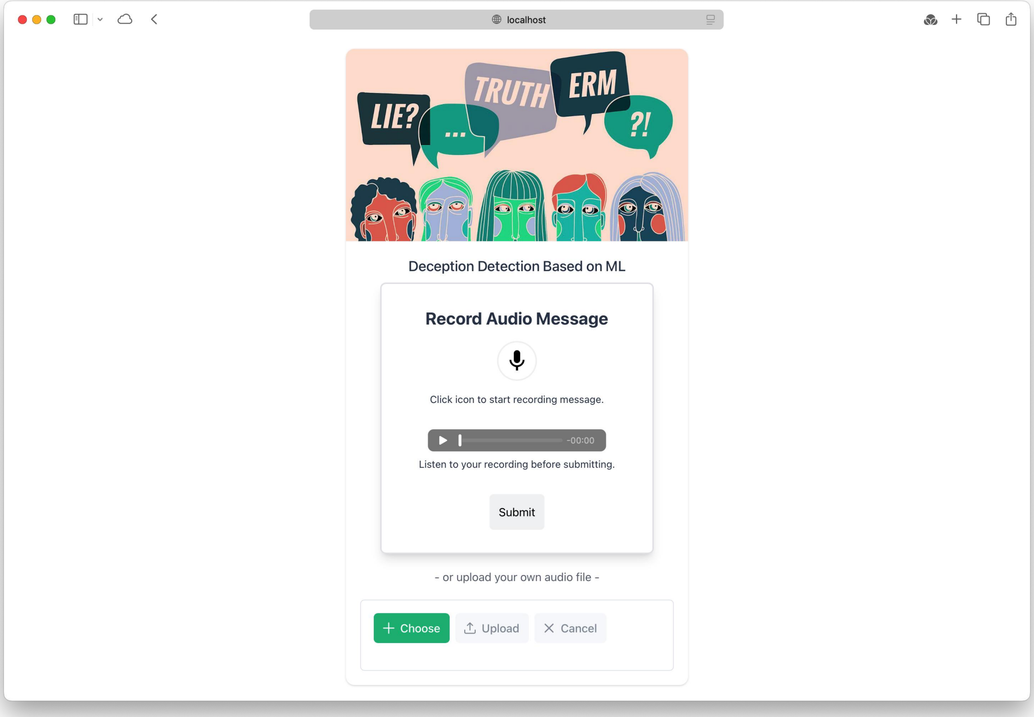
PrimeVue is a UI component library for Vue.js that provides a wide array of ready-made components designed to enhance the speed and quality of front-end development. It is highly customizable and suitable for both simple and complex applications. It also allows easy integration with Vue's Composition API. We used its FileUpload component for uploading audio files and the ConfirmDialog component for displaying results in our web application.

Axios

Axios is a popular JavaScript library used for making HTTP requests from both the browser and Node.js environments. It provides an easy-to-use API for sending asynchronous requests (GET, POST, PUT, DELETE, etc.) to interact with RESTful

APIs. Axios supports features like request and response interceptors, automatic JSON transformation, and error handling. It is adopted in our system to achieve communication between front-end and back-end servers.

8.2.2Application Design



Deception Detection Based on ML

Record Audio Message

Click icon to start recording message.

Listen to your recording before submitting.

- or upload your own audio file -

Figure 8-1: Web Application Page Interface

Our web application is quite simple and easy to use. It supports both sound recording using the device microphone and audio file uploading. The user can click on the microphone icon in the upper frame to start recording and click once more to stop. The audio will be saved for playback before submitting. This recording module comes from an open-source Vue component called vue-audio-tapir. Or by simply clicking the "Choose" button in the lower frame, the user can upload their own audio file in a variety range of formats, which will be converted into a WAV file before being sent to the back-end server. After choosing either one of the approaches, the machine-learning model in the back-end will generate an outcome with a probability attached, and then the result is displayed on the web page.

8.3 Back-End Development

8.3.1 System Architecture

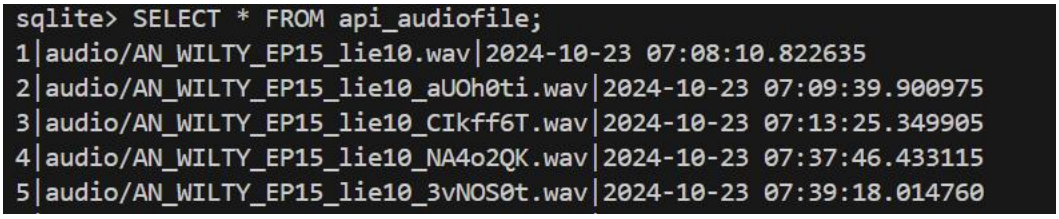
The back-end is designed using the Django framework, which adopts the MVC design pattern (Model-View-Controller), which effectively reduces the coupling between modules and facilitates code refactoring afterwards.

8.3.2 Database Design

The system defines an audio model to manage the audio files uploaded by users. The model definition contains the file uploaded by the user and the corresponding upload time, which will facilitate the further management of the file.

In the database, the model corresponds to a table where the primary key is a unique identifier while other fields are used to store the audio file path and upload time. These fields allow the system to efficiently manage and retrieve audio data.

For example, when the user uploads an audio file, the system will generate a record in the database, including the path of the audio file and its upload time.



```
sqlite> SELECT * FROM api_audiofile;
1|audio/AN_WILTY_EP15_lie10.wav|2024-10-23 07:08:10.822635
2|audio/AN_WILTY_EP15_lie10_aUOh0ti.wav|2024-10-23 07:09:39.900975
3|audio/AN_WILTY_EP15_lie10_CIkff6T.wav|2024-10-23 07:13:25.349905
4|audio/AN_WILTY_EP15_lie10_NA4o2QK.wav|2024-10-23 07:37:46.433115
5|audio/AN_WILTY_EP15_lie10_3vNOS0t.wav|2024-10-23 07:39:18.014760
```

Figure 8-2: Example of Audio File Upload Record in Database

8.3.3 Api Design

The API design of this project includes three main modules: Serializer, URL, and View. These modules together constitute the complete process of audio file processing, ensuring efficient and reliable data exchange between the front-end and back-end. The following figure shows the relationship and data flow between these three modules.

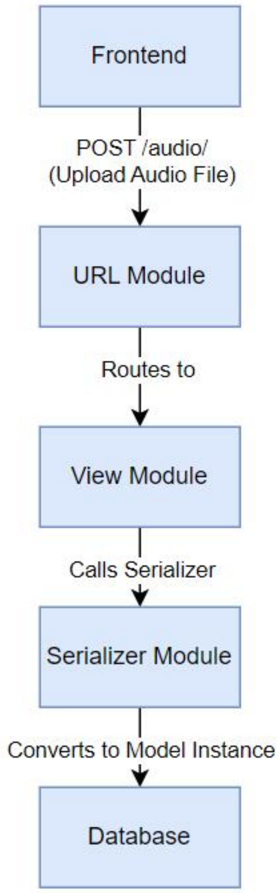


Figure 8-3: API Design

8.3.3.a Serializer Module

In this module AudioFileSerializer is defined as the serializer. Its function is the serialization and deserialization of data. When the API needs to return audio file information, the deserializer automatically converts the model data to JSON format by defining fields such as id, audio_file, and uploaded_at. While when users upload audio files through the API, the serializer validates the input data .After validation it converts the data into a model instance which is then saved to the database.

8.3.3.b URL Module

This module mainly defines the routing of audio file related APIs, and uses the routing function of Django REST Framework in the process of implementation, which makes the management and access of APIs more convenient and effective.

The module sets up an audio folder for storing audio files uploaded by users. The module sets up the audio folder for storing the audio files uploaded by users and includes the URL patterns generated by the router in the list of urlpatterns. When a user accesses the URL of an audio file, it will be routed to the corresponding module to handle the request.

8.3.3.c View Module

This module is responsible for implementing the processing of the user's incoming audio and returning the corresponding results.

When receiving the creation request, the module will call serializer to validate the uploaded data. If the validation is successful, the system saves the audio file to the database and generates the corresponding record. After the uploaded audio file is loaded, the module will convert the data accordingly and use the trained model to make a prediction, after which the prediction result will be sent back to the front-end.

9. Challenges and Future Work

9.1 Obstacles in System Development

In the system development phase, two main areas of difficulty were encountered. The first of these was the the limitation of data and features during the data processing module. The second was the integration of the reasoning module, with a particular focus on optimising the accuracy rate and the interpretation.We will describe these problems in detail and give possible solutions in ***section 9.2.***

9.1.1 Limitation of Data and Features

Despite the extraction of two gigabytes of audio files, comprising 1,684 pieces of audio, the resulting dataset remains relatively modest in scale for a personal computer. Further data is required to support the training of the reasoning engine, but this is challenging for the team at this juncture. Furthermore, in regard to the extraction of features, it should be noted that there are numerous alternative approaches, including the use of Harmonic-to-Noise Ratio and Linear Predictive Coding, among others. However, due to the limited resources of the equipment and other factors, we only performed the MFCC extraction. While the majority of the sound's information was captured, a minor portion was not.

9.1.2 Problem for the accuracy and generation of reasoning engine

Although the Soft Voting Integration Model_A demonstrated a high level of accuracy on the test set, it remains unclear whether this accuracy can be generalized to other datasets, as well as to environments with significant disturbances (e.g., noise) and specific environments where the user's voice may be unstable (e.g., interrogation). It is plausible that prisoners may become nervous during interrogation. It is also conceivable that the model may be susceptible to being deceived by sophisticated lying techniques. Therefore, relying on audio information alone, it remains challenging to achieve a high degree of accuracy in determining whether the user (the perpetrator) is lying or not. The current results can be used as a point of reference, but only as such.

9.2 Strategies to Overcome Identified Challenges

9.2.1 Methods to solve limitation of Data and Features

The dearth of datasets and extracted features can be addressed through the continuous acquisition of new users' voice judgments, which are then incorporated into the knowledge base following some form of canonical processing and assigned appropriate labels. This approach not only addresses the issue of data scarcity but also ensures the inclusion of a diverse range of voices, which is crucial for the extraction of more comprehensive data. It is hoped that more sophisticated equipment or enhanced graphics cards will facilitate the acceleration of the training process.

9.2.2 Methods to solve problem for the accuracy and generation

In order to address the issues identified in the preceding section, it is necessary to obtain further information, including data pertaining to facial expressions and body movements. This will facilitate the continued development of a multimodal engine based on the integrated model, thereby enhancing the reasoning engine's capacity for comprehensive and robust judgement. Furthermore, the incorporation of additional data input can facilitate the enhancement of this capability.

9.3 Additional Features to be Implemented

In the future, the following features may be added to enhance the functionality of the site.

The functionality for user registration and login must be incorporated into the system so that users can register and log in. Once logged in, registered users are able to view and manage their audio files. Conversely, unregistered users are unable to log in, thus ensuring the protection of user privacy and data security.

To facilitate data comprehension, the system can present the user's audio analysis results and historical data in graphical form. To illustrate, the system displays the trend of analysis results for disparate audio files. Intuitive data visualisation enables users to comprehend the analysis results and extract pertinent information.

Furthermore, the system incorporates an audio file management function. This allows users to view a list of their uploaded audio files, including the corresponding upload date, file size, and other pertinent information in the front-end interface. For each audio file, users can elect to delete or download the file, thus facilitating more effective management.

References

[1] Xiu, Noé, et al. "Lie Detection Based on Acoustic Analysis." *Journal of Voice* (2024).

[2] Talaat, F.M. Explainable Enhanced Recurrent Neural Network for Lie Detection Using Voice Stress Analysis. *Multimedia Tools and Applications,* vol. 83, 32277–32299 (2024).

[3] Xue, Alice, Hannah Rohde, and P. A. Finkelstein. "An Acoustic Automated Lie Detector." Princeton University (2019).

[4] Inbau, Fred, Joseph Buckley, and Brian Jayne. *Criminal Interrogation and Confessions*. Jones & Bartlett Publishers, 2013.

[5] Alicex2020. (2019). Deep Learning Lie Detection. GitHub repository. https://github.com/alicex2020/Deep-Learning-Lie-Detection.

[6] Huang, G.-L. (2021). Deception Detection. GitHub repository. https://github.com/come880412/Deception_detection.

[7] Yang, F. (2022). Lie Detection. GitHub repository. https://github.com/yyf20001230/Lie_Detection.

[8] Curci, Antonietta, et al. "Accuracy, Confidence, and Experiential Criteria for Lie Detection Through a Videotaped Interview." *Frontiers in Psychiatry*, vol. 9, 2019, p. 748.

[9] Xi, Ji, et al. "A Semi-Supervised Lie Detection Algorithm Based on Integrating Multiple Speech Emotional Features." *Applied Sciences*, vol. 14, no. 16, 2024, p. 7391.

[10] Mendels, Gideon, et al. "Hybrid Acoustic-Lexical Deep Learning Approach for Deception Detection." *Proceedings of Interspeech*, 2017.

[11] Bareeda, EP Fathima, BS Shajee Mohan, and KV Ahammed Muneer. "Lie detection

using speech processing techniques." *Journal of Physics: Conference Series*. vol. 1921. no. 1, 2021.

[12] Al-Dhaher, Fadi K., Duraid Y. Mohammed, and Mohammed Khalaf. "The Most Important Features of Lie Detection Using Voice Stress." *Al-Iraqia Journal for Scientific Engineering Research*, vol. 3, no. 1, 2024, pp. 93-110.

[13] Fernandes, Sinead V., and Muhammad S. Ullah. "Use of machine learning for deception detection from spectral and cepstral features of speech signals." *IEEE Access*, vol. 9, 2021, pp. 78925-78935.

[14] Guo, X., Selvaraj, N.M., Yu, Z., Kong, A., Shen, B., and Kot, A. "Audio-Visual Deception Detection: DOLOS Dataset and Parameter-Efficient Crossmodal Learning." *International Conference on Computer Vision (ICCV)*, 2023.

[15] Selvaraj, Nithish. "Audio-Visual-Deception-Detection-DOLOS-Dataset-and-Parameter-Efficient-Crossmodal-Learning." 2023, GitHub, https://github.com/NMS05/Audio-Visual-Deception-Detection-DOLOS-Dataset-and-Parameter-Efficient-Crossmodal-Learning.

[16] Tiwari, Vibha. "MFCC and Its Applications in Speaker Recognition." *International Journal on Emerging Technologies*, vol. 1, no. 1, 2010, pp. 19-22.

[17] Gupta, Shikha, et al. "Feature Extraction Using MFCC." *Signal & Image Processing: An International Journal*, vol. 4, no. 4, 2013, pp. 101-108.

[18] Kumari, Saloni, Deepika Kumar, and Mamta Mittal. "An Ensemble Approach for Classification and Prediction of Diabetes Mellitus Using Soft Voting Classifier." *International Journal of Cognitive Computing in Engineering*, vol. 2, 2021, pp. 40-46.

[19] Rigatti, Steven J. "Random Forest." *Journal of Insurance Medicine*, vol. 47, no. 1, 2017,

pp. 31-39.

[20] Cortes, Corinna, and Vladimir Vapnik. “Support-Vector Networks.” *Machine Learning*, vol. 20, no. 3, 1995, pp. 273-297.

[21] Cover, Thomas M., and Peter E. Hart. “Nearest Neighbor Pattern Classification.” *IEEE Transactions on Information Theory*, vol. 13, no. 1, 1967, pp. 21-27

[22] Hinton, Geoffrey E., et al. “A Fast Learning Algorithm for Deep Belief Nets.” *Neural Computation*, vol. 18, no. 7, 2006, pp. 1527-1554.