A Real-Time Voice Fraud Detection Framework using Whisper and Fine-Tuned Phi-3 Mini for Context-Aware Fraud Classification

Adekemi O. Amoo
Department of Software Engineering, Obafemi Awolowo University, Ile-Ife, Nigeria

Damilare E. Bakare
MosaiQ Labs, London, United Kingdom

Theresa O. Omodunbi
Department of Information System, Obafemi Awolowo University, Ile-Ife, Nigeria

Mary T. Onifade
Department of Software Engineering, Obafemi Awolowo University, Ile-Ife, Nigeria

Samuel I. Omilo
Department of Computer science & Cybersecurity, Obafemi Awolowo University, Ile-Ife, Nigeria

ABSTRACT
Voice fraud has recently become a critical cybersecurity concern, where fraudsters impersonate trusted individuals over phone calls to deceive victims and obtain confidential information. Conventional fraud detection systems, which rely on predetermined rules or post-call detection, are increasingly becoming ineffective against evolving fraud patterns and do not ensure user safety in real-time.

This study proposes the design and implementation of a real-time voice fraud detection system that analyses real-time conversations to identify potential fraudulent activity. The proposed system integrates OpenAI's Whisper model for speech-to-text transcription with a fine-tuned Phi-3-mini Large Language Model (LLM) for contextual fraud detection. The system architecture uses the Browser MediaStream API for audio capture and WebSocket communication to transmit streaming data to a FastAPI-based backend for real-time processing.

The fine-tuned LLM was trained on a dataset comprising 447 labeled conversational samples and subsequently evaluated on 307 unseen test samples, achieving an overall accuracy of 95%. Experimental results demonstrate a precision of 97% and a recall of 98% for normal conversation, and a precision of 75% and a recall of 67% for fraudulent conversation. These findings underscore the effectiveness of leveraging context-aware language modelling and real-time audio transcription in detecting fraudulent speech patterns, thereby surpassing the limitations of traditional rule-based systems.

The study contributes to the body of knowledge by demonstrating the feasibility of combining speech processing and large language models to enable proactive, real-time detection of voice fraud during phone calls.

Keywords
Audio Stream Processing, Deep Learning, Large Language Models (LLMs), Real-Time Fraud Detection, Speech-to-Text Processing.

1. INTRODUCTION
Voice communication remains a fundamental medium through which businesses and government agencies interact with their customers or citizens. The global Voice over Internet Protocol (VoIP) market has experienced remarkable expansion, valued at over USD 116 million in 2024 and projected to reach over USD 176 million by 2029 [13]. However, this rapid adoption has also created new avenues for malicious exploitation. Fraudsters increasingly impersonate legitimate businesses or government officials to exploit users' trust, tricking them into disclosing sensitive information. Such actors often employ concepts like social engineering techniques to deceive victims who are unaware of the ongoing manipulation [14].

In 2024, the U.S Federal Trade Commission (FTC) reported over USD 12 billion lost by consumers to fraud, with a major proportion of these losses attributed to imposter scams conducted through phone calls [5]. This alarming trend exposes a critical vulnerability in voice-based communication systems: the inherent credibility of real-time conversation allows fraudsters to assume false identities and manipulate victims into revealing their sensitive details.

To mitigate these risks, many organizations have invested in user education initiatives, teaching their customers how to verify the legitimacy of communications. Despite these efforts, user awareness remains insufficient, and these schemes continue to evolve. Consequently, fraud detection systems that rely on post-call analysis are no longer adequate, as sensitive information is often compromised during the call itself. Detecting fraud in real-time has therefore become imperative.

Previous research works have explored rule-based, statistical(e.g., Naive Bayes method), and post-call transcript analysis approaches to voice fraud detection. While partially effective, these methods are limited in many ways. Rule-based systems depend on static keywords or metadata patterns, thus struggle to adapt to new and evolving fraud strategies.

Given these limitations, there is a pressing need for a proactive and adaptive fraud detection framework capable of analyzing ongoing conversations in real time. Recent advancements in Natural Language Processing (NLP) and Large Language Models (LLMs) demonstrate the potential to understand conversational context and detect subtle linguistic cues indicative of deception.

This study proposes the development of a real-time voice fraud detection system designed to analyze live spoken dialogues and identify fraudulent activity during ongoing calls. The proposed system aims to overcome the limitations of post-call detection methods by combining real-time speech-to-text conversion and

context-aware language modeling, thereby enhancing the security and reliability of voice communication channels.

2. LITERATURE REVIEW

Generally, fraud call tactics are deceptive schemes intended to exploit the vulnerability of individuals or organisations. The deception can range from pretending to be your bank to selling you goods or services that are fake. The focus of it all is to obtain money or information from you under the pretense [1]. Some of the common tactics utilised in fraud calls include impersonation, spoofing, digital information manipulation, and social engineering.

Social engineering stands out as the most commonly employed tactic. It involves manipulating individuals into divulging confidential information or performing actions that compromise their security. Rather than exploiting technical vulnerabilities, social engineering targets human psychology by invoking urgency, fear, or trust to make it difficult for people to identify fraud [14].

Before the popularity of intelligent systems, fraud detection in voice communications relied on simpler, more static techniques. These methods were often rule-based and manual, suitable only for defined fraud patterns [14].

i. Manual review and user complaints: Fraud detection largely depended on human intervention, mostly through direct complaints to customer support or security staff. This approach, while valuable in confirming fraud, was slow and lacked scalability.

ii. Rule-based detection: Systems were configured with fixed rules, such as flagging calls from certain regions, unusual call times, or rapid call bursts. They rely on existing rules, making them less adaptive [3]. Although straightforward, these rules often fail to detect novel or evolving fraud strategies due to a lack of adaptability

iii. Caller blacklists: This is popularly used around the world. Known fraudulent numbers are added to a list and automatically blocked or flagged. Users can report fraudulent numbers for the database to be updated. This method required constant updates and offered little protection against first-time offenders or spoofed numbers.

iv. Basic pattern recognition using metadata Basic heuristics such as call frequency, duration, and time of day were used to identify suspicious behavior [17]. These patterns could hint at fraud but lacked contextual awareness, leading to many false positives or missed threats.

The real-time aspect of fraud detection is important, particularly in voice communication, in order to provide immediate protection to users [16]. Real-time processing integrates well with VAD to ensure only relevant audio is processed, and analysis happens in gaps between statements.

Deep Learning models have been applied widely in fraud detection in spoken or transcribed conversations. These models help detect suspicious patterns in the content and sequence of speech or text. Depending on the task complexity and available features, models can be chosen for their speed, accuracy, or interpretability.

Logistic regression is a basic model for binary classification, offering a straightforward way to separate fraudulent from non-fraudulent calls. It is a technique for assessing the likelihood of a binary result determined by a number of reasonable factors [7]. The simplicity and interpretability of logistic regression make it suitable for early-stage detection systems.

SVMs (Support Vector Machines) are effective in high-dimensional text spaces and perform well on smaller, labeled datasets. With proper feature engineering (e.g., TF-IDF or word embeddings), they can classify text-based transcripts with strong margins as used by [7] in their study.

Large Language Models (LLMs) are some of the most advanced tools currently available for understanding and generating human language. These models are trained on massive datasets and are able to interpret not just words, but also the context, tone, and intent behind them [9]. In the context of fraud detection, this makes LLMs especially powerful.

Unlike earlier fraud detection models that depend heavily on predefined features or keyword spotting, LLMs can understand full conversations. They are able to pick up on subtle cues, understand sentences in context, and identify conversations that might signal fraudulent intent [14]. Because they are general-purpose, they can also be adapted for different use cases with relatively little data through fine-tuning or prompting.

[14] used deep learning architectures, including Convolutional Neural Networks (CNNs), Long Short-Term Memory (LSTM) networks, and Stacked Denoising Autoencoders (SDAEs), to detect fraudulent patterns in structured telecom datasets. Their hybrid model achieved above 99% accuracy, showing the potential of deep learning in automated feature extraction. However, their method was confined to offline analysis of numerical data, lacking contextual understanding of spoken interactions.

[1] created NaLie, a mobile fraud detection app using Natural Language Processing (NLP) and Crowd Machine Learning (CrowdML) for SMS-based financial scams. Their model used Naïve Bayes and Decision Trees to classify user-reported text messages, continuously improving through crowd feedback. Despite its effectiveness in text analysis, the model did not apply to voice-based communication or real-time voice fraud.

[16] also developed a voice fraud detection system that converted speech into text using an Automatic Speech Recognition (ASR) engine and classified it with a Naïve Bayes model. Their approach relied on keyword-based probabilistic classification using TF-IDF features extracted from transcribed calls. The system achieved promising accuracy in identifying fraudulent communication but lacked semantic depth, making it ineffective in recognizing context-dependent manipulation or evolving fraud tactics. The same year, [3] applied machine learning models, specifically Support Vector Machines (SVMs) and Recurrent Neural Networks (RNNs), to detect anomalies in Call Detail Records (CDRs) based on parameters like call duration, frequency, and time. Their model reached over 90% accuracy, yet it analyzed only metadata, not the conversational content. This limitation restricted its ability to detect fraud embedded in seemingly normal call behavior.

Lastly, [14] introduced a Retrieval-Augmented Generation (RAG)-based Large Language Model (LLM) for real-time text-based fraud detection. The system combined an LLM with a dynamic knowledge retrieval module to recognize evolving scam patterns and policies. Although it achieved 97.98% accuracy, it focused solely on text data and was not integrated with real-time audio processing or speech recognition.

Overall, the reviewed works demonstrate significant advancements in fraud detection through statistical, deep learning, and LLM-based techniques. Yet, the common limitation across these studies is the absence of real-time, speech-integrated contextual analysis. This study proposes a system gap that combines Whisper-based live transcription

with a fine-tuned Phi-3-mini LLM to achieve real-time voice fraud detection during real-time voice calls.

3. METHODOLOGY

The development of the proposed real-time voice fraud detection systems uses an iterative approach, which is guided by well-defined system requirements. The Unified Modelling Language (UML) tool, specifically the conceptual diagram, the use case diagram, the sequence diagram, and the Flow Diagram, was used to model and design the system. For the implementation, the system was developed using Python (FastAPI framework), the WebSocket protocol for real-time communication, and the Browser MediaStream API for live audio capturing and streaming. These technologies ensured low-latency and real-time interaction between the client and the backend server.

3.1 Requirement Gathering

Requirement gathering is an important part of building the voice fraud detection system. It helps define what the system needs to work properly and meet the objectives set earlier in the project. This includes understanding the tools, data, and processes required to detect fraudulent voice interactions, especially those involving impersonation or financial scams.

According to a 2025 report by the U.S. Federal Trade Commission (FTC), consumers lost $12.5 billion to fraud in 2024, with $2.95 billion coming from imposter scams—many of which were carried out through phone calls. This highlights the need for systems that can detect fraud during voice conversations and help reduce these losses.

With this in mind, the requirements gathered focus on making the system capable of listening to voice input, converting it to text, analyzing for suspicious patterns, and providing timely alerts. These requirements are grouped into functional and non-functional categories.

1. Functional requirements:

Audio Input and Processing: The system should be able to receive and process live or recorded voice calls.

Speech-to-Text Conversion: A reliable system is needed to transcribe speech accurately.

Fraud Pattern Detection: The system should analyze the transcript to detect signs of fraud, such as urgency, impersonation, or requests for sensitive information.

Alert System: When fraud is suspected, the system should raise an alert in real time or after analysis.

2 Non-functional requirements

Accuracy: The system must be accurate in detecting fraud to avoid false positives and negatives.

Speed: It should process data quickly enough for real-time or near-real-time use.

Scalability: The system should be able to handle multiple users without performance issues.

Security and Privacy: Any recorded or processed data must be handled securely to protect user privacy.

3.2 Data Collection and Preparation

The data used for developing the voice fraud detection system was sourced from two main categories: open-source transcription datasets and a conversational analysis study focused on human interaction patterns. The goal was to gather a mix of both genuine and potentially fraudulent conversations that could help train and evaluate the system.

Part of the data was collected from a conversational analysis of scam calls study by [1], which explored naturally occurring conversations. The study included data from people who voluntarily recorded their phone conversations, as well as interactions gathered from public social media platforms. These conversations were transcribed and analyzed to highlight patterns in speech behavior, especially how individuals respond to requests, express urgency, or seek clarification.

In addition to that, open-source transcription datasets from platforms like Kaggle were used to supplement the data. These datasets provided a broad range of transcribed speech, which helped in making the system more generalizable.

3.3 Conceptual Diagram

The conceptual diagram in Fig. 1 describes the workflow of the proposed voice fraud detection system. The user initiates and streams live audio, which is sent in small chunks to the WebSocket client server. Each audio chunk is forwarded to the Speech-to-Text server for automatic transcription, and the resulting text is passed immediately to the fine-tuned LLM for contextual fraud analysis. The LLM classifies each statement as normal or fraudulent and returns the result to the WebSocket server, which then displays the feedback to the user while the conversation is still ongoing.

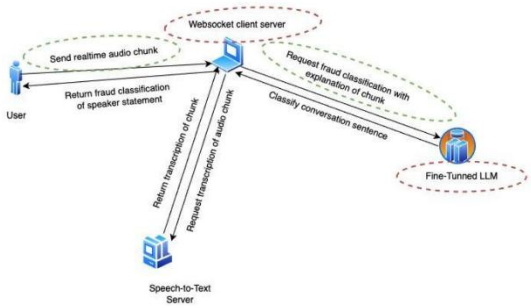


Fig. 1: Conceptual Diagram of the System.

3.4 Sequence Diagram

The sequence diagram, as shown in Fig. 2, shows the chronological order of interaction between the system modules. The sequence typically begins with an audio input, followed by transcription, then fraud analysis, and finally an alert if any fraud is detected.

3.5 Flow Diagram

The Flow Diagram provides a step-by-step operational view of the entire system. It begins with audio acquisition, followed by chunking and transmission, transcription, context analysis, and alert generation. Decision nodes specify conditional branching, where each utterance is labeled as Normal or Fraudulent. This diagram demonstrates the closed control loop that enables continuous fraud monitoring during an active voice call. The flow diagram in Fig. 3 captures the entire pipeline from tuning to analysis.

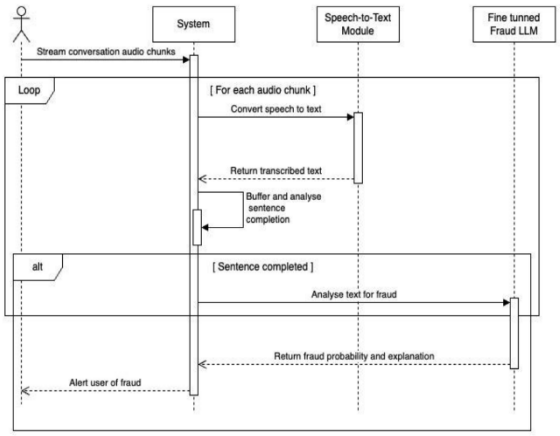


Fig. 2: Sequence diagram of the real-time call fraud analysis system

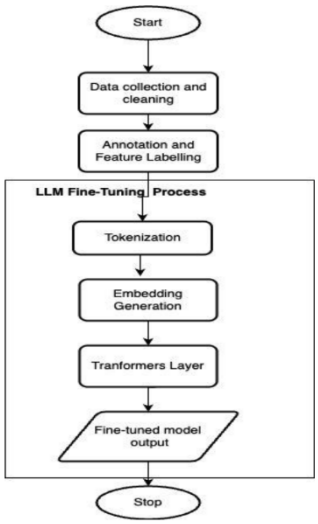


Fig. 3: Flow diagram of the call fraud analysis system

3.6 Use Case Diagram

The use case diagram shows the various ways actors can interact with the system. It highlights the main functions the system supports and the actors involved in each case. In this system, key use cases include: submitting an audio recording for analysis, viewing the audio classification results, transcribing recordings, analyzing transcribed data for fraudulent patterns, and labeling data for fine-tuning. The use case diagram is shown in Fig. 4

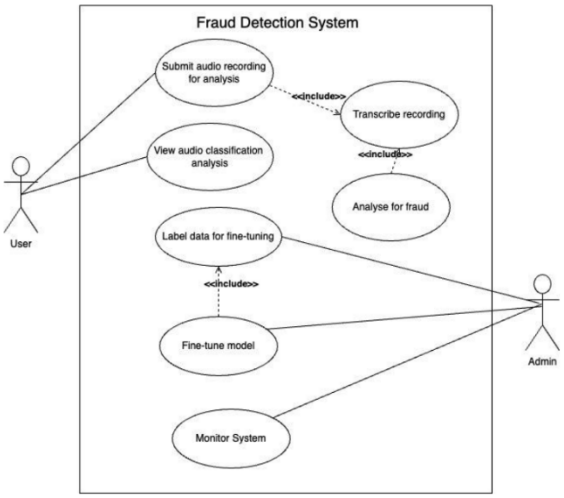


Fig. 4: Use case diagram of the call fraud analysis system

4. SYSTEM IMPLEMENTATION

The implementation of the real-time voice fraud detection system integrates web-based audio streaming, automatic speech recognition, and a fine-tuned language model to detect fraudulent intent during live voice conversations. The system comprises a browser client for audio capture, a FastAPI backend for real-time processing, Whisper for speech transcription, and a fine-tuned LLM for fraud detection.

4.1 Audio Processing

Audio is captured directly from the user's microphone using the Browser MediaStream API and ReactJS. The stream is segmented into small audio chunks and transmitted to the backend via a WebSocket connection, enabling continuous, low-latency communication. On the backend, the audio is buffered and processed by OpenAI Whisper, which converts each completed speech segment into text. Whisper's robustness to noise and accents ensures reliable transcript quality in real-world voice call scenarios.

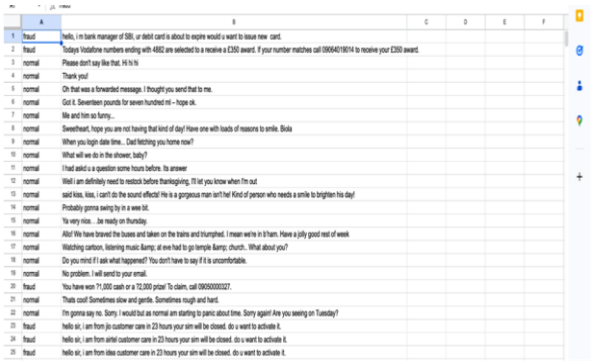
4.2 Model Fine-Tuning and Training

The fraud detection model is based on the Phi-3-mini LLM, which has been fine-tuned on 447 manually labeled statements representing both normal and fraudulent speech. Fine-tuning was carried out using Google Colab and Hugging Face Transformers, enabling the model to learn contextual fraud cues such as impersonation language, urgency, emotional pressure, and attempts to extract sensitive information. The trained model was later deployed locally using Ollama to support low-latency inference.

The call statements data used for training and evaluation were collected from Kaggle. The dataset contained call statements and their fraud classification, as shown in Figure 5. This dataset was cleaned and formatted into instruction-style prompts used for fine-tuning and evaluation of the model, as shown in Fig. 6.

4.3 Statement Identification Algorithm

To enable real-time decision-making, the system includes a statement identification algorithm. The Voice Activity Detection (VAD) routine determines when a full spoken statement has ended. Once an end-of-utterance boundary is detected, the corresponding audio segment is passed for transcription and analysis. This ensures that fraud classification occurs incrementally during the call rather than after the call ends.



4.4 Fraud Analysis

Each transcribed statement is submitted to the fine-tuned LLM for classification. The model analyzes context and intent rather than only keywords, returning one of two outcomes: a normal statement or a potentially fraudulent statement. Alongside the classification label, the model generates a short explanation to justify the decision, enhancing interpretability and user trust. This happens on the WebSocket backend server. Fig, 7 contains the implementation of this algorithm.

4.5 Tools

i. Google Colab: Google Colab is a cloud-based notebook environment used for training, testing, and running experiments without the need for a powerful local machine. It provided free access to GPUs and TPUs, which made it easier to accelerate model training.

ii. Python FastAPI: FastAPI served as the backend framework for building the fraud detection system's API. I chose it because of its speed, simplicity, and support for asynchronous operations. Its support for WebSockets was crucial for implementing real-time audio processing, allowing the system to capture live call audio, transcribe it, and run fraud analysis with low latency.

iii. ReactJS: For the frontend, ReactJS was used to create a dynamic and responsive user interface. This allowed end-users to interact with the fraud detection system seamlessly, from uploading audio files to monitoring live transcription and fraud alerts. React's component-based architecture was used to keep the code modular and maintainable. The frontend is also integrated with the FastAPI backend via REST and WebSocket connections, ensuring smooth real-time communication between the client and server.

iv. OpenAI Whisper: Whisper was the core speech-to-text model that transcribed phone calls into text. Its robustness to accents, background noise, and low-quality audio made it particularly useful for handling real-world call scenarios. The transcribed text became the foundation for fraud detection tasks.

v. Hugging Face: Hugging Face was the platform used to access, fine-tune, and manage large language models for fraud detection. It provided pre-trained models and tools like the Transformers library, which accelerated development and experimentation. By fine-tuning models on fraud-related call data, general-purpose LLMs can be adapted into domain-specific detectors of fraudulent language patterns.

vi. Ollama: Ollama was used to run the fine-tuned fraud detection model locally, making it possible to test the system without relying on cloud-based inference. This setup reduced latency and dependency on external services.

4.6 User Interface

The user interface was developed using ReactJS and provides controls to start and stop monitoring, manage microphone access, and display live transcripts with fraud indicators. The interface communicates with FastAPI via REST API for backend interactions and WebSockets for real-time audio and model output. Transcribed statements are displayed immediately, with fraudulent ones highlighted and accompanied by the explanation generated by the LLM. The results, including transcription output and fraud risk assessment, are presented in a structured and readable format. To enhance usability, the UI uses a component-based design that separates functionalities such as audio capture, transcription display, fraud alert visualization, and system notifications. Figure 8 illustrates the fraud analysis output for a normal statement, while Figure 9 demonstrates the system's response to a fraudulent statement.

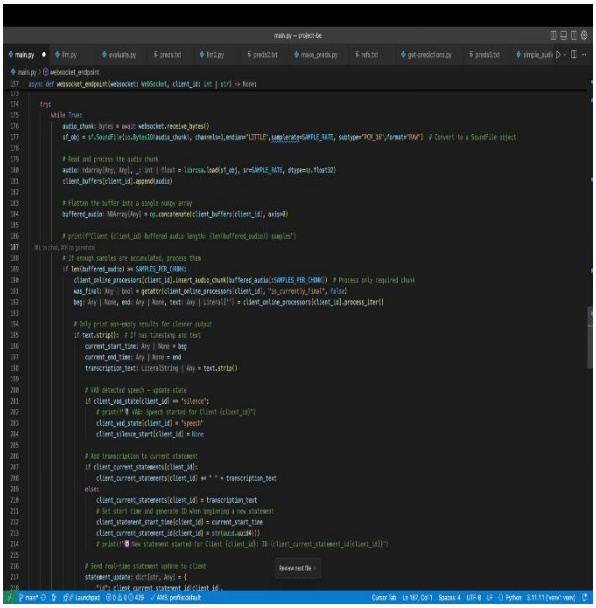
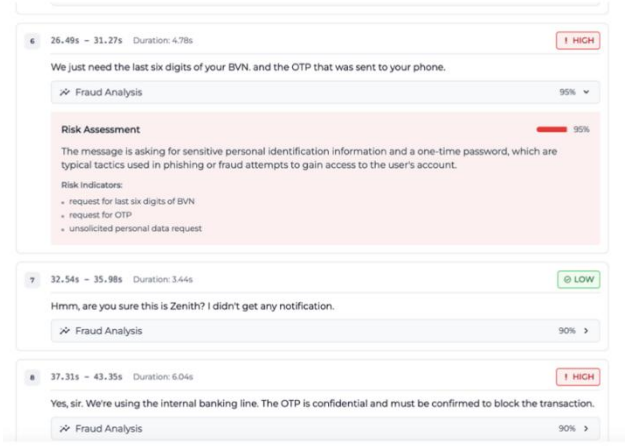


Fig. 7: Websocket Endpoint Implementation



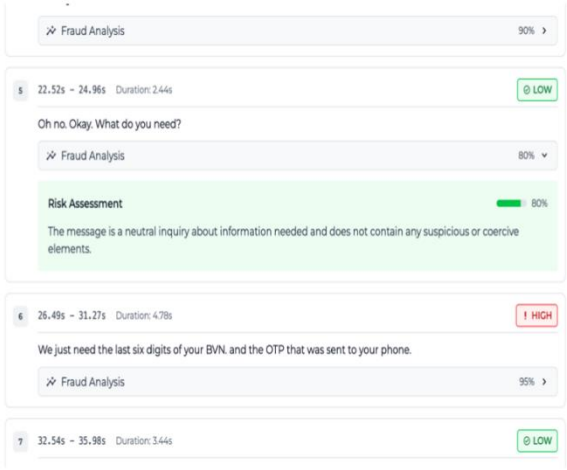


Fig. 8: Fraud Analysis of a Normal Statement

Fig. 9: Fraud Analysis of a Fraudulent Statement

5. SYSTEM EVALUATION

The performance of the real-time voice fraud detection system was evaluated using unseen test data consisting of 307 conversational statements, and the computed results are shown in Figure 10 below. The results indicate that the model effectively distinguishes between fraudulent and normal speech, achieving an overall accuracy of 95%.

For the fraud class, the model attained a precision of 0.75, meaning that three out of every four statements flagged as fraudulent were correctly classified. The recall score of 0.67 indicates that approximately two-thirds of all truly fraudulent statements were successfully detected, while some cases remained unidentified. The F1-score of 0.71 demonstrates a moderate balance between precision and recall, showing that although the model is capable of recognising fraud, further optimisation could improve its ability to capture all fraud attempts.

The normal class exhibited consistently strong performance, with a precision of 0.97, a recall of 0.98, and an F1-score of 0.97. These results confirm that the system is highly reliable at identifying legitimate statements and generates very few false alarms during normal voice interactions.

The macro-averaged metrics (precision = 0.86, recall = 0.82, F1-score = 0.84) show balanced performance across both classes without favouring one category over the other. Meanwhile, the weighted averages (precision = 0.95, recall = 0.95, F1-score = 0.95) reinforce the robustness of the system, accounting for the class imbalance in the dataset (27 fraudulent statements vs. 280 normal statements).

Overall, the evaluation demonstrates that the proposed system is highly effective in detecting voice fraud in real time, with outstanding performance on normal speech and strong recognition of fraudulent language patterns. Figure 11 presents the confusion matrix summarising the distribution of correct and incorrect predictions for both classes.

```
===== Scikit-Learn Evaluation =====
num_examples: 307
accuracy: 0.9511
precision_micro: 0.9511
recall_micro: 0.9511
f1_micro: 0.9511
precision_macro: 0.8591
recall_macro: 0.8226
f1_macro: 0.8396

===== Detailed Classification Report =====
              precision    recall  f1-score   support

       fraud       0.75      0.67      0.71        27
      normal       0.97      0.98      0.97       280

    accuracy                           0.95       307
   macro avg       0.86      0.82      0.84       307
weighted avg       0.95      0.95      0.95       307
```

Fig. 10: Evaluation Results from Scikit-Learn

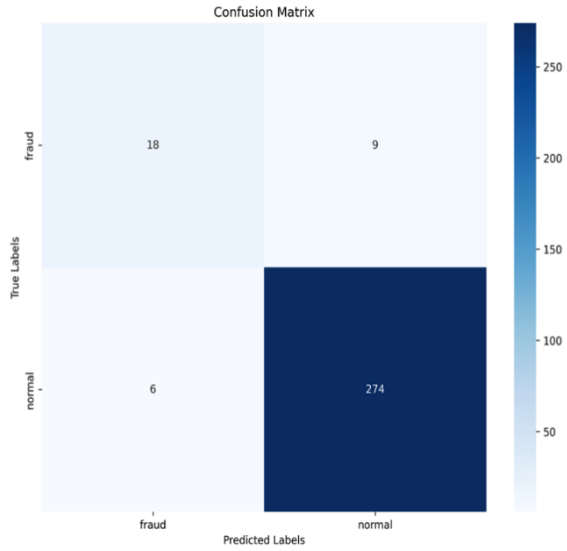


Fig. 11: Confusion Matrix of Evaluation

6. LIMITATIONS

Although the system demonstrates strong performance in detecting fraudulent speech in real time, several limitations were identified.

1. The detection model relies heavily on the quality of transcription provided by the Whisper ASR; inaccurate transcription caused by poor audio quality may affect downstream fraud classification. 2. While the model performs well with normal statements, its recall for fraudulent statements indicates that some fraud attempts may still go undetected, especially when scammers use subtle or unfamiliar manipulation tactics that are not well represented in the training dataset.

3. Real-time processing requires stable network connectivity; interruptions in WebSocket streaming can delay transcription and classification feedback.

7. SUMMARY, CONCLUSION, AND FUTURE WORK

7.1 Summary

This project set out to address the rising threat of voice fraud, which continues to grow as fraudsters find new ways to exploit phone conversations. Existing fraud detection methods are often rule-based or limited to spotting keywords, which makes them rigid and easy to bypass once attackers change their approach. To address this problem, this study develops a system that uses a fine-tuned Large Language Model (LLM) to detect fraud in real time, with a focus on understanding the context of conversations rather than just surface-level cues.

The system was built around a pipeline that connects several key components. Audio is captured in the browser using the MediaStream API, streamed to the backend via WebSockets, and converted into text using OpenAI Whisper. From there, a fine-tuned Phi-3-mini model is used to classify statements as either fraudulent or normal. Hugging Face was used to manage and fine-tune the model, while Ollama enabled running it locally for testing and privacy. On the infrastructureside, the backend was built with FastAPI, handling both REST and WebSocket communication, and the frontend was developed in ReactJS to provide users with a real-time, interactive interface.

To measure performance, the system was evaluated on a separate dataset of 307 samples not used during training. Using the scikit-learn library, the model achieved 95% accuracy, with excellent performance on normal statements and solid results on fraud detection. These findings show that the approach is not only feasible but also practical, with clear potential for real-world application in environments like banking and customer support.

7.2 Conclusion

This research presented a real-time voice fraud detection system that integrates automatic speech recognition with a fine-tuned large language model to detect fraudulent intent during conversations. The system successfully demonstrated that combining live audio streaming, low-latency transcription, and contextual language analysis enables proactive fraud detection rather than post-call response. Evaluation results showed an overall accuracy of 95%, with excellent performance on normal statements and strong performance on fraudulent statements, confirming the system's feasibility and effectiveness for real-time deployment.

7.3 Future Work

While the system has shown strong results, there are a few areas that can be improved in future work. One of the key findings from the evaluation was that, although the model performed well overall, its recall for fraudulent statements was lower compared to normal statements. This means that some fraud cases were missed. To address this, future efforts should focus on expanding the training dataset with a wider variety of fraud examples. Having more representative samples will help the model recognize fraud more reliably, even when the patterns are subtle or unfamiliar.

Another important step would be to test the system in real-world environments. The experiments so far were conducted under controlled conditions, but actual call centers and financial institutions present different challenges, such as heavy background noise, diverse accents, and even code-switching between languages. Deploying and evaluating the system in such environments would provide a stronger measure of its robustness and adaptability.

It would also be valuable to connect the system with existing fraud management tools. In practice, fraud detection rarely works in isolation. Integrating this system with transaction monitoring, customer verification, or alert systems would create a more complete defense mechanism.

Adding a feedback mechanism where users can confirm or dispute flagged statements would also improve the system over time and raise awareness of fraud tactics. This kind of interaction would allow the model to learn continuously while also keeping users informed and alert to new fraud strategies.

8. REFERENCES

[1] Ifeoma, I.-I. A., and Anulika, C. O. 2022. Conversational analysis of scam calls in Nigeria. Awka Journal of Linguistics and Languages, Special Edition 2, pp. 269–297.

[2] Behera, S. K., and Nayak, M. M. 2020. Natural language processing for text and speech processing: A review paper. SSRN Scholarly Paper No. 3878634. Social Science Research Network. Available at: https://papers.ssrn.com/abstract=3878634

[3] Bhargavi, D. K., and Shivani, B. M. 2024. Detection of fraudulent phone calls in mobile applications. Turkish Journal of Computer and Mathematics Education (TURCOMAT), 15(2).

[4] Detecting fraud calls vis-à-vis natural language processing. n.d. ResearchGate. Available at: https://www.researchgate.net/publication/381320394_Detecting_Fraud_Calls_vis-a-vis_Natural_Language_Processing

[5] Federal Trade Commission. 2025. New FTC data show a big jump in reported losses to fraud to $12.5 billion in 2024. March 2025. Available at:

https://www.ftc.gov/news-events/news/press-releases/2025/03/new-ftc-data-show-big-jump-reported-losses-fraud-125-billion-2024 [Accessed 22 April 2025].

[6] Hong, B., Connie, T., and Goh, M. K. O. 2023. Scam calls detection using machine learning approaches. In Proceedings of the 11th International Conference on Information and Communication Technology (ICoICT), pp. 442–447.

[7] Hossain, M. I. n.d. Software development life cycle (SDLC) methodologies for information systems project management.

[8] IBM Think. 2024. What is NLP (natural language processing)? 11 August 2024. Available at: https://www.ibm.com/think/topics/natural-language-processing [Accessed 9 June 2025].

[9] Jiang, L. 2024. Detecting scams using large language models. arXiv. DOI: 10.48550/arXiv.2402.03147.

[10] Macháček, D., Dabre, R., and Bojar, O. 2023. Turning Whisper into a real-time transcription system. In Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics: System Demonstrations, pp. 17–24. DOI: 10.18653/v1/2023.ijcnlp-demo.3.

[11] Miah, M. S. U., Kabir, M. M., Sarwar, T. B., Safran, M., Alfarhood, S., and Mridha, M. F. 2024. A multimodal approach to cross-lingual sentiment analysis with an

ensemble of transformer and LLM. DOI: 10.1038/s41598-024-60210-7.

[12] Olorunshola, O. E., and Ogwueleka, F. N. 2022. Review of system development life cycle (SDLC) models for effective application delivery. In Joshi, A., Mahmud, M., Ragel, R. G., and Thakur, N. V. (Eds.), *Information and Communication Technology for Competitive Strategies (ICTCS 2020)*, Vol. 191, pp. 281–289.

[13] RapidBTS. 2024. VoIP growth in Africa: Market size and forecast. Available at: https://rapidbts.ng/voip-growth-africa/ [Accessed 22 February 2025].

[14] Singh, G., Singh, P., and Singh, M. 2025. Advanced real-time fraud detection using RAG-based LLMs. Version 1. *arXiv*. DOI: 10.48550/ARXIV.2501.15290.

[15] Subudhi, S., and Panigrahi, S. 2018. A hybrid mobile call fraud detection model using optimized fuzzy C-means clustering and group method of data handling-based network. *Vietnam Journal of Computer Science*, 5(3), pp. 205–217. DOI: 10.1007/s40595-018-0116-x.

[16] Valarmathi, C., and Sharanya, S. 2024. Scam call detection using NLP and Naïve Bayes classifier. *International Journal of Scientific Research in Engineering and Management*, 8(07), pp. 1–6.

[17] Xing, J., Yu, M., Wang, S., Zhang, Y., and Ding, Y. 2020. Automated fraudulent phone call recognition through deep learning. *Wireless Communications and Mobile Computing*, 2020, pp. 1–9.

[18] Zhan, T., Shi, C., Shi, Y., Li, H., and Lin, Y. 2024. Optimization techniques for sentiment analysis based on LLM (GPT-3). *arXiv*. DOI: 10.48550/arXiv.2405.09770.

[19] Zhao, Q., Chen, K., Li, T., Yang, Y., and Wang, X. 2018. Detecting telecommunication fraud by understanding the contents of a call. *Cybersecurity*, 1(1), p. 8.