A Real-Time AI-based Framework for Vishing Detection - Guarding Conversations: A Real-Time AI-based Framework for Vishing Detection on iOS

Shizhe Jia[1][0009−0006−0952−4988], Alexander Nußbaum[2][0009−0001−0008−4756], and Florian Alt[1,2][0000−0001−8354−2195]

[1] Ludwig Maximilian University of Munich, Munich, Germany
Shizhe.Jia@campus.lmu.de
florian.alt@ifi.lmu.de
[2] University of the Bundeswehr Munich, Munich, Germany
alexander.nussbaum@unibw.de

Abstract. Voice phishing (vishing) is an increasingly common threat in which attackers exploit phone conversations to extract sensitive information. Existing defenses—such as blacklists and caller ID authentication—struggle against spoofing and cannot assess the content of live calls. We present a real-time vishing detection framework for iOS that transforms a virtual assistant into a passive in-call security agent. The system transcribes conversations, combines keyword spotting with LLM–based semantic analysis, and delivers both immediate alerts and post-call reports. In an evaluation across 96 real and synthetic calls, the framework achieved 99.9% keyword detection accuracy, 91.7% semantic accuracy, and near real-time responsiveness. Performance was robust across accents, genders, and noisy environments. Our findings demonstrate the feasibility of content-driven vishing defense and highlight how intelligent voice assistants can support secure, usable human–computer interaction.

Keywords: Vishing · Phishing Detection · Voice Assistant · Real-time Security · Speech Recognition · Large Language Models · Human-Computer Interaction

1 Introduction

Voice-based communication technologies have become integral to modern life, with virtual assistants embedded in smartphones, smart speakers, and IoT devices enabling users to interact with digital services in more natural ways [1, 8, 27]. At the same time, generative AI and deepfake technologies have significantly exacerbated the threat of vishing attacks, in which fraudsters exploit voice calls to deceive users into revealing sensitive information, a problem likely to intensify as the prevalence of voice-based interfaces increases [3, 6, 9]. Unlike email or text-based phishing, vishing eliminates visual cues, making it harder for victims to detect deceit and resist psychological manipulation [2, 23].

Current defenses like blacklists and STIR/SHAKEN focus on caller identification and are ineffective against spoofing or content-based deception [19, 29].

In this work, we investigate whether a virtual voice assistant—equipped with automatic speech recognition (ASR) and large language model (LLM) capabilities—can serve as a viable, passive defense against vishing. Our goal is to address a tangible cybersecurity threat affecting consumers, enterprises, and vulnerable populations, while ensuring that added protection does not disrupt usability in everyday voice interactions.

To this end, we designed, implemented, and evaluated a real-time vishing detection system for iOS devices. The system transforms a virtual assistant into an in-call security agent: it transcribes conversations with Whisper ASR, highlights high-risk keywords, and forwards transcript segments to GPT-3.5 Turbo for semantic analysis. When suspicious content is detected, the assistant provides immediate in-call alerts and generates post-call reports that summarize risk levels and explain why particular segments appeared threatening. Unlike network-level defenses, our approach analyzes spoken content rather than caller metadata, making it resilient against spoofing. Furthermore, the system integrates personalized post-call feedback, turning detection into a mechanism for raising long-term phishing awareness, and its modular design supports future extensions such as multilingual deployment or offline functionality.

We evaluated the framework across 96 phone call scenarios, including both real-world vishing attempts and controlled synthetic conversations. The system achieved 99.9% keyword detection accuracy and 91.7% semantic classification accuracy, with alerts delivered on average within 8.7 seconds—fast enough for near real-time operation. Performance remained robust across accents, genders, and noise conditions, and the LLM reliably surfaced phishing intent even in complex social engineering cases. These findings demonstrate that virtual assistants can be repurposed from passive responders into active security agents, filling a critical gap in the vishing defense landscape. More broadly, our work contributes to secure human–computer interaction by showing how AI-powered conversational monitoring can enhance both safety and user trust in voice interfaces.

Contribution Statement. This paper makes three contributions: (1) We present a real-time, content-based vishing detection framework for iOS that leverages Whisper ASR, GPT-3.5 Turbo, and keyword spotting to transform virtual assistants into in-call security agents. (2) We report on the results of a study of 96 real and synthetic calls showing that the system achieves high accuracy with robustness across accents, genders, and noise conditions. (3) We synthesize recommendations for how intelligent voice assistants can unobtrusively support user safety, including balancing immediacy of alerts with minimal disruption, providing explanatory post-call feedback to sustain awareness, and designing for extensibility (e.g., multilingual support, offline operation).

2 Background and Related Work

2.1 Acoustic (Voice) Phishing – A Growing Concern

Voice phishing, or *vishing*, refers to social engineering attacks exploiting voice communication to deceive victims into revealing sensitive information [3, 6, 25]. Thereby, adversaries typically impersonate trusted entities such as banks, government agencies, or technical support through phone calls, voicemails, or Voice over Internet Protocol (VoIP) services [9, 30]. By exploiting psychological mechanisms such as trust, authority, urgency, and fear, attackers persuade users to disclose credentials, banking details, or other private data [2]. The absence of visual or contextual cues, which are often present in text- or email-based phishing, further increases user susceptibility during spoken interactions [14, 15, 23, 24].

Recent developments have intensified the threat: Automated *robocalls* with caller ID spoofing reached billions annually [3], while *audio deepfakes* enable highly convincing voice impersonations [4, 28]. Virtual assistants (e.g., Siri, Alexa) further increase vulnerability, as users trust auditory interfaces more and scrutinize spoken content less [15, 24]. Vishing succeeds by exploiting cognitive biases, such as the truth-default theory, where victims assume honesty until strong evidence suggests otherwise [2].

2.2 Existing Countermeasures and Their Limitations

Current defenses operate primarily at the network or user level:

- **Blacklists** block known fraudulent numbers using crowdsourced or regulatory data (e.g., Truecaller, FTC) [19, 20]. They achieve moderate effectiveness (50–70%) but are reactive and easily bypassed via spoofing [6, 10, 20].
- **STIR/SHAKEN** introduces digital signatures for VoIP calls to verify caller identity [3, 18, 29]. While reducing spoofing in the U.S., it remains limited to IP networks, incompatible with legacy telephony infrastructure (Public Switched Telephone Network, PSTN), and ineffective against international calls or verified-number abuse [11, 26, 29].
- **Awareness training** (in-person, gamified, or embedded) improves detection but suffers from rapid decay—effectiveness often returns to baseline within months [5, 13, 22].

Table 1 summarizes these approaches. Collectively, they fail to analyze *conversational content in real time*, leaving users vulnerable during ongoing calls where deception unfolds gradually. This gap motivates content-based, real-time defenses integrated into everyday voice interfaces, such as virtual assistants equipped with ASR and LLM capabilities.

2.3 Comparison with Recent Content-based Approaches

Several recent works have explored machine learning for vishing detection, primarily in Korean-language contexts or on Android platforms.

Table 1. Comparison of common vishing countermeasures

Method	Strengths	Limitations	Scope
Blacklists	Easy deployment	Reactive; spoofing bypass	Pre-call blocking
STIR/SHAKEN	Prevents spoofing	U.S.-limited; no content check	Caller authentication
Training	Improves awareness	Short-term effect	User education

Table 2. Comparison with recent content-based vishing detection approaches.

System	Detection	Platform	Real-time	Language	LLM
Lee & Park [17]	Content (ML)	Android	Yes	Korean	No
HearMeOut [16]	App behavior	Android	Yes	Any	No
Park et al. [21]	User study	—	—	—	—
Cimino et al. [10]	Content (LLM)	Web	Partial	English	Yes
Ours	**Content (ASR+LLM)**	**iOS**	**Yes**	**English**	**Yes**

Lee and Park [17] proposed a real-time detection system using basic ML models (e.g., SVM, Random Forest) on transcribed Korean call content, focusing on rapid training with named entity recognition and n-grams (i.e., contiguous sequences of n words used as text classification features). While effective for scripted scams, their approach relies on traditional classifiers without leveraging modern large language models (LLMs) for deeper semantic understanding.

In contrast, Kim et al. [16] developed HearMeOut, an Android-specific framework that detects malicious app behaviors enabling vishing (e.g., call redirection, overlay attacks, synthetic voice playback) through runtime API monitoring. This system targets app-level threats rather than conversational content and requires no transcription or semantic analysis.

Park et al. [21] conducted a user study on perceptions of on-device AI vishing detection apps, revealing privacy concerns (e.g., feeling "wiretapped") despite local processing; however, their work is qualitative and does not propose a technical detection system.

Table 2 provides a structured comparison. Our framework differs by combining robust ASR (Whisper) with LLM-based (GPT-3.5 Turbo) semantic analysis for real-time, content-driven detection on iOS, operating passively during live calls without relying on app behavior monitoring or being limited to specific languages.

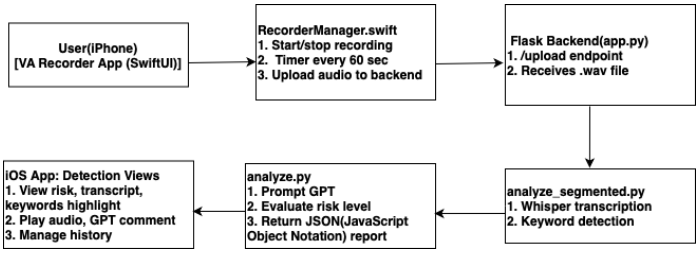

Fig. 1. System architecture. The iOS frontend (left) handles recording and user interaction via SwiftUI. The Flask backend (right) performs Whisper transcription, keyword detection, and GPT-3.5 Turbo risk analysis, returning structured JSON reports.

2.4 Research Gap

While prior work addresses caller identification and prevention, few solutions monitor ongoing call content [19]. Notably, no existing system performs real-time, content-based vishing detection during live conversations using modern ASR and LLM capabilities on consumer devices. This work bridges this gap by leveraging automatic speech recognition (ASR) and large language models (LLMs) to detect manipulation cues directly from spoken dialogue.

Building on this gap, the following section details our system's design and implementation, directly addressing the need for real-time, content-based analysis through ASR and LLM integration.

3 System Design and Implementation

We developed an iOS-based framework that transforms a virtual assistant into a passive, real-time vishing detection agent. iOS was chosen due to its strict privacy model, unified audio access during calls, and widespread adoption among consumers. The system operates unobtrusively during live phone calls, transcribes speech, analyzes content for phishing indicators, and provides immediate alerts and post-call reports.

3.1 Overview and Architecture

The system integrates native iOS call audio capture with a lightweight Flask backend for heavy computation (Figure 1). During an active call, audio is recorded in 60-second segments using AVFoundation, transcribed locally or on the server with OpenAI's Whisper ASR, and analyzed via keyword spotting and GPT-3.5 Turbo for semantic risk assessment. Alerts are displayed instantly on the device, and detailed reports are generated post-call.

The user workflow (Figure 2) is seamless: the user answers calls normally while the assistant runs in the background. Upon detection of suspicious content, a non-disruptive text alert appears. After the call, a report summarizes risks, highlights flagged phrases, and includes LLM-generated explanations.

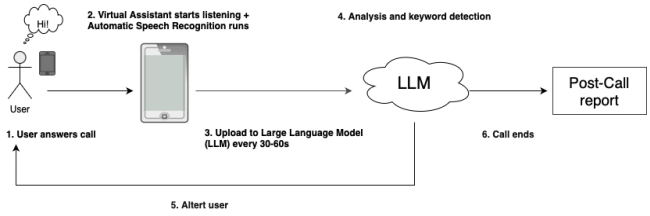

Fig. 2. User workflow during a monitored call. The assistant passively captures audio, uploads 60-second segments for ASR and LLM analysis, alerts the user in real time upon risk detection, and generates a post-call report when the call ends.

Table 3. Threat scope and detection methods.

Scope	Threat Type	Detection Method
In-scope	Mass campaigns, human-led social engineering, caller ID spoofing	Keyword spotting + LLM semantic analysis
Out-of-scope	Highly targeted attacks, adversarial LLM-generated speech	—

3.2 Threat Model

The system targets common vishing attacks relying on linguistic cues (Table 3). It is resilient to spoofing since detection is content-based. Evasion via phrase rewording or delayed requests is partially mitigated by the LLM's contextual understanding.

3.3 Design Goals

The system was guided by the following key design goals, motivated by the real-world constraints of live phone conversations and user privacy expectations:

Real-time detection: Alerts must be delivered with low latency (target <10 seconds end-to-end) to enable users to react before disclosing sensitive information. This is achieved through segmented processing (60-second audio windows), efficient local transcription, and lightweight LLM queries.

Unobtrusive operation: The assistant operates passively in the background without interrupting or revealing its presence to the caller. Alerts are silent, text-based notifications, and no audio feedback is provided during the call to preserve natural conversation flow.

Privacy preservation: Sensitive audio and transcripts must remain under user control. Raw audio is processed and stored exclusively on-device (or on a controlled local server), with automatic deletion options. Transcription uses a local Whisper implementation to avoid sending audio to third-party clouds, while post-call reports are encrypted and user-manageable.

Robustness: The system must perform reliably across diverse real-world conditions, including varying accents, speaker genders, and background noise. This is addressed by selecting state-of-the-art ASR (Whisper) and evaluating performance systematically across these dimensions (see Section 4).

Extensibility: The modular architecture supports future enhancements such as full offline operation, multilingual support, integration of newer LLMs, or hybrid defenses combining content analysis with network-level signals.

3.4 Key Components

The system combines complementary detection mechanisms:

Automatic Speech Recognition (ASR): Whisper was selected for its high accuracy across accents and noise, and compatibility with downstream OpenAI services. Transcription runs locally where possible to support privacy.

Keyword Detection: A curated database of over 40 high-risk phrases (e.g., "verification code", "urgent transfer", "account suspended") is scanned in real time. The initial keyword list was compiled by prompting ChatGPT to generate terms commonly associated with vishing across thematic domains (urgency cues, identity verification, financial solicitation, and authority impersonation), then validated against published social engineering taxonomies [2, 25]. The list is dynamically updatable to accommodate emerging attacks.

Semantic Analysis: GPT-3.5 Turbo receives rolling 30–60 second transcript segments plus detected keywords and classifies risk using a structured prompt. The prompt instructs the model to act as a phishing risk analyst:

"You are a phishing risk analyst. A transcript of a voice call is provided below. Transcript: [transcript]. Keywords detected: [keyword list]. Based on tone, urgency, manipulation tactics, and suspicious content, briefly describe in 1–2 sentences whether the conversation shows signs of phishing."

GPT-3.5 Turbo was chosen over smaller on-device language models because it provides strong zero-shot classification across diverse social engineering patterns without requiring task-specific fine-tuning or curated vishing training data [7]. While smaller models could reduce API dependency, they would require substantial labeled datasets for comparable semantic performance—a direction we identify as future work (Section 5).

Alerting: Risk is categorized as Low/Medium/High based on keyword count and LLM output. Alerts are silent, text-based pop-ups to avoid disrupting the conversation.

Post-Call Reports: Each analyzed segment generates a local report with transcript, highlighted keywords, risk score, LLM explanation, and optional audio playback for user review.

3.5 Implementation Details

The prototype consists of a SwiftUI-based iOS frontend and a Python/Flask backend.

Frontend (iOS) Uses AVFoundation for audio recording, URLSession for segment upload, and SwiftUI for interface (status indicators, alerts, history view). Reports are stored locally using Codable and FileManager.

Backend (Flask) Receives audio via a secure upload endpoint, runs Whisper transcription, performs keyword matching, queries GPT-3.5 Turbo, and returns structured JSON risk reports. All processing is modular for easy extension.

The implementation prioritizes low latency (average 8.7 seconds end-to-end, see Section 4), privacy (local storage, no persistent cloud logging), and extensibility (e.g., future offline models, multilingual support).

Whisper was chosen for its state-of-the-art robustness to accents and noise [12], while GPT-3.5 Turbo offers an optimal balance of speed, cost, and accuracy for real-time semantic analysis [7].[3]

The iOS frontend provides a simple and intuitive user interface (see Figures 3 and 4). Figure 3 shows the main recording screen, where users can manually start and stop monitoring with clearly labeled buttons (green "Start" and red "Stop") and view real-time status information. During an active call, potential phishing risks are communicated via non-intrusive pop-up notifications (Figure 4). These alerts display the detected risk level (e.g., "High Risk"), list flagged keywords, and include a brief explanatory message, allowing users to quickly assess the situation without disrupting the conversation.

4 Evaluation

4.1 Evaluation Objectives

We evaluated the system's technical performance and real-world feasibility using 96 phone calls (88 synthetic + 8 real-world), systematically varying accent (British/American), gender (male/female), and background noise (quiet/noisy). This section assesses how well the system meets our design goals, focusing on accuracy, responsiveness, and robustness.

The evaluation targeted five dimensions linked to the design goals:

ASR accuracy: Whisper's transcription fidelity across accents, genders, and noise.
Keyword detection: Precision in identifying phishing cues.
LLM reliability: GPT-3.5 Turbo's semantic classification accuracy.
System responsiveness: End-to-end latency for real-time alerts.
Robustness: Resilience to acoustic variability.

Tests were conducted on an iPhone 16 Pro simulator with a stable network (upload ∼32 Mbps). Synthetic speech was generated using NaturalReader TTS[4] with controlled accents and genders; noise was added at moderate TV volume. The dataset comprised 52 phishing and 44 benign calls (see Tables 5, 4 and 6 for details). A curated keyword database of 40+ phrases supported detection.

[3] The source code will be made publicly available on GitHub upon paper acceptance.
[4] https://www.naturalreaders.com/

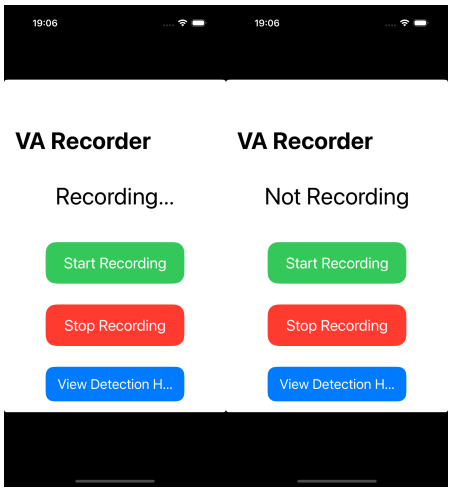

Fig. 3. Recording UI States: during recording, not recording

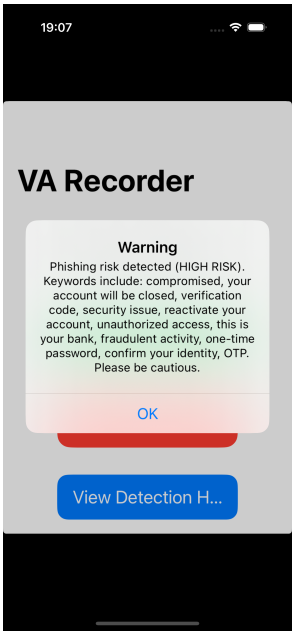

Fig. 4. Alert Popup Displayed After Detection

Table 4. Summary of evaluation dataset.

Type	Phishing	Benign
Synthetic (controlled variables)	48	40
Real-world	4	4
Total	52	44

Study Design We used a within-subjects factorial design to test robustness. Independent variables: accent (British/American), gender (male/female), noise (quiet/noisy TV at vol. 20). Each was paired with 11 dialogue scripts (Table 5) for 88 synthetic calls, plus 8 real-world (Table 6).

Dependent variables measured ASR/keyword/LLM accuracy, alert correctness (low/medium/high vs. ground truth), and latency.

4.2 Results

ASR Recognition Accuracy We evaluated the transcription reliability of the Whisper ASR engine under realistic conditions by examining how accuracy varied across the independent variables—*accent* (British vs. American), *speaker gender* (male vs. female), and *background noise* (quiet vs. noisy environment). Transcription quality was quantified using the Normalized Generalized Leven-

Table 5. Synthetic dialogue dataset used for controlled evaluation.

Call ID	Caller Identity / Scenario	Phishing
A	None (unsolicited caller)	Yes
B	Friend	No
C	Impersonated Friend	Yes
D	Family Member	No
E	Impersonated Mother	Yes
F	Company Colleague	No
G	Impersonated IT Support	Yes
H	Bank Representative	No
I	Impersonated Bank	Yes
J	Embassy Official	No
K	Impersonated Embassy	Yes

Table 6. Real-world voice calls used in the evaluation.

Call ID	Phishing	Description
L	Yes	Fraudster claiming to be a police officer.
M	Yes	Credit card scam voicemail.
N	Yes	Caller impersonating law enforcement.
O	Yes	Voicemail requesting immediate payment.
P–S	No	Casual conversations between friends.

shtein Distance (NGLD) [31], which provides a consistent, length-independent measure of similarity between spoken and transcribed text (cf. Table 7).

Main effects. Accent had a small influence: British speech (96.6%) slightly outperformed American (94.7%), possibly because synthetic British voices were slower and more clearly articulated. Gender produced the largest gap: female voices (96.8%) exceeded male (94.5%), consistent with prior findings that higher-pitched, uniformly articulated voices are easier for ASR systems to segment. Noise caused a modest decline from 96.5% (quiet) to 94.8% (noisy), demonstrating Whisper's robustness in low-to-moderate noise environments.

Interaction effects. Under clean conditions, all groups exceeded 96.8%, but compounding factors degraded performance for American male voices in noise (90.1%)—a compound vulnerability when lower pitch, faster pacing, and background interference coincide. Female voices were remarkably stable across all conditions (British: 96.8%/96.8%; American: 96.9%/96.7% quiet/noisy).

Intermediate Summary. Whisper demonstrated consistently high performance, with only minor declines under specific condition combinations. These findings emphasize the importance of accounting for user diversity and environmental variability when designing voice-based security systems.

Table 7. ASR accuracy (%) by experimental condition (main effects; sel. interactions).

Variable	Condition	Accuracy (%)
Accent	British / American	96.6 / 94.7
Gender	Female / Male	96.8 / 94.5
Noise	Quiet / Noisy	96.5 / 94.8
Best: British female (quiet)		96.8
Worst: American male (noisy)		90.1

Transcription and Keyword Detection Accuracy To assess the reliability of speech processing and keyword identification, we analyzed both the transcription accuracy of Whisper and the precision of the system's rule-based keyword detection module. These dependent variables capture the system's ability to process natural speech and extract relevant linguistic cues indicative of phishing activity across varying acoustic conditions.

Across all *96 calls* (52 phishing, 44 benign), Whisper achieved a high average transcription accuracy of *97.8%*, demonstrating robustness across the independent variables of accent, speaker gender, and background noise. This level of accuracy indicates that the ASR component provides sufficiently precise input for downstream semantic analysis.

The keyword detection module also performed exceptionally well, reaching *99.9% accuracy* in identifying predefined phishing-related terms. These results confirm that the combined pipeline of ASR and keyword matching can reliably detect explicit phishing cues in both synthetic and real-world calls. However, *91.7% of all calls* were categorized as *low risk*, suggesting that the static keyword database may not fully capture more nuanced or context-dependent strategies.

Intermediate Summary. Together, these findings underscore the importance of maintaining a dynamic and extensible keyword database that evolves alongside emerging phishing tactics. While high accuracy on explicit cues reflects strong technical performance, future iterations should incorporate adaptive keyword updates or contextual learning mechanisms to improve sensitivity to subtler indicators of manipulation.

GPT Judgement Accuracy and Robustness To evaluate the reliability of the system's semantic classification, we analyzed how consistently the GPT-3.5 Turbo module identified phishing intent across the 96 calls. Every 60 seconds, the model received transcribed dialogue segments with contextual keywords and was prompted to assign a phishing risk level. This measure reflects the system's ability to interpret conversational cues—such as tone, urgency, or requests for financial information—and distinguish manipulative from legitimate communication under varying conditions.

Overall classification accuracy reached *91.7%* across all test calls. The model achieved *100% accuracy* on the eight real-world recordings, successfully detecting authentic scams such as law enforcement impersonation, credit card fraud,

and urgent payment requests. Accuracy was slightly lower for synthetic dialogues (*90.1%*), largely due to one false positive case in which a legitimate bank representative discussed account verification—a scenario where strong lexical cues outweighed benign conversational context. Importantly, the model's performance remained stable across the independent variables of accent, speaker gender, and background noise, consistently recognizing patterns of urgency, authority, and financial solicitation that characterize social engineering.

Intermediate Summary. These results demonstrate that GPT-3.5 Turbo provides a reliable semantic layer for real-time phishing detection, effectively identifying conversational manipulation across diverse conditions. Nonetheless, the occurrence of isolated false positives showcases a limitation of context-sensitive interpretation, particularly in cases involving legitimate financial or service-related dialogue. Future iterations could incorporate refined prompt design or contextual calibration to improve disambiguation, reducing alerts while remaining sensitive to genuine threats.

System Response Time To assess real-time performance, we measured end-to-end latency from segment completion to alert display, covering file saving, upload, Whisper transcription, GPT-3.5 analysis, and frontend rendering.

Across 20 test iterations, the average response time was *8.7 seconds*, with a range between *5.4* and *10.6 seconds*. We consider this sufficient for near real-time vishing mitigation because social engineering calls typically unfold over several minutes of rapport-building before the critical disclosure request [2, 23]; a sub-10-second alert after each 60-second segment thus provides warning well before most victims would share sensitive information. The pipeline remained stable throughout testing, and no significant latency fluctuations were observed across different accents, genders, or noise conditions. A minor edge case occurred when calls ended mid-upload, temporarily interrupting analysis; this issue was addressed through the implementation of a short buffer mechanism that ensured complete data transfer before processing.

Intermediate Summary. The sub-10-second latency demonstrates readiness for deployment in delay-sensitive scenarios. By maintaining fast processing while performing complex ASR and LLM inference, the system achieves a practical balance between analytical depth and responsiveness, supporting real-time user intervention during high-pressure phishing attacks.

5 Limitations and Future Directions

As a proof-of-concept, this work prioritized technical feasibility and system validation over exhaustive generalization. The proposed framework demonstrates strong performance in transcription, keyword detection, semantic classification, and responsiveness, yet several *limitations* remain. Rather than undermining the contribution, these aspects define the natural boundaries of this initial exploration and highlight promising directions for future research.

A first limitation concerns the *data basis*. Of the 96 evaluated calls, only eight were real recordings, while the remainder were generated using ChatGPT scripts and NaturalReader TTS. While 96 calls constitute a limited sample, the controlled factorial design ($2 \times 2 \times 2 \times 11$ scenarios) ensures systematic coverage of key variability dimensions, and results were consistent across all conditions. Nonetheless, synthetic voices lack the nuances of natural prosody, hesitations, and emotional tone. Moreover, the current scenario set focuses on impersonation-based scams and does not yet cover emerging vishing vectors such as fake job offers, fraudulent insurance assistance, or targeted appeals to specific demographics (e.g., international students). Future work should expand to larger and more diverse real-world datasets with broader scenario coverage.

A second limitation relates to the *static keyword database*. While the curated list achieved 99.9% accuracy under controlled conditions, it cannot automatically adapt to evolving phishing language—for instance, novel pretexts such as offering free legal assistance or health insurance. This constraint is inherent to rule-based detection and emphasizes the need for *continual learning mechanisms* or user-driven updates. Systematically expanding keywords could leverage crowdsourced reporting, periodic LLM-based extraction from documented scam transcripts, and user feedback loops.

The *semantic classification component* also presents challenges. The GPT-3.5 Turbo module reached an overall accuracy of 91.7%, but occasional false positives occurred when legitimate financial or security-related conversations were misclassified as phishing. Although this conservative bias is acceptable for a safety-critical prototype, future work should focus on refining contextual modeling to balance sensitivity and specificity. Leveraging additional cues—such as caller identity or dialogue intent—could help disambiguate legitimate exchanges from manipulative ones. Furthermore, the current reliance on a cloud-based LLM introduces API dependency; future iterations could explore *fine-tuned smaller language models* (SLMs) for on-device inference, which would reduce latency and improve privacy at the cost of requiring curated vishing-specific training data.

A related issue arises from *prompt dependency*: model performance depends on the structure and clarity of prompts, and changes in future LLM versions could affect consistency. This motivates future research into prompt optimization and automated calibration. Similarly, *latency and computational cost* impose constraints—the 8.7-second average response time under stable broadband may increase on weaker networks, and Whisper's computational demand may limit offline deployment. Future *hybrid architectures* combining on-device and cloud inference could address both performance and privacy.

Beyond technical factors, further improvements are needed in *system stability and usability*. Occasional edge cases occurred when calls ended mid-upload, temporarily interrupting analysis. Although a buffer mechanism was introduced to mitigate this issue, more robust session handling and recovery mechanisms will be required for deployment in uncontrolled settings. Similarly, this study did not yet include a *usability evaluation*. As our focus was on technical feasibility, aspects such as trust, alert comprehension, and distraction risk remain

unexplored and should be addressed through future user studies to assess how real-time alerts influence user attention and behavior during calls.

Finally, broader considerations of *privacy, multilinguality, and hardware performance* must be addressed. Although all audio data were processed locally and encrypted during transmission, real-world deployment will require formal compliance with data protection frameworks such as GDPR and transparent user consent procedures. Moreover, the system currently supports English only. Extending it to other languages will necessitate retraining or fine-tuning to account for linguistic and cultural variation. Importantly, all tests were conducted on an iPhone 16 Pro *simulator*; real-device deployment may encounter additional constraints such as iOS sandboxing restrictions on background audio capture during calls, microphone access limitations, battery consumption under continuous ASR processing, and App Store review policies. These platform-specific challenges must be addressed through real-device testing before practical deployment.

Taken together, these limitations define the natural evolution points of a first technical demonstration and highlight clear opportunities for advancing the adaptability, scalability, and human-centered design of real-time, AI-assisted vishing detection.

6 Implications for Design

Our study demonstrates the feasibility of transforming virtual voice assistants into real-time, in-call security agents. Beyond technical validation, the results highlight opportunities for designing intelligent assistants that promote user security without compromising usability or trust.

6.1 Unobtrusive In-Call Support

Designing for real-time intervention requires carefully balancing immediacy and usability. In our system, alerts were delivered as brief, text-based notifications rather than audio prompts, allowing users to remain focused on the ongoing call while still being informed of potential risks. This finding underscores a broader design principle: *security feedback in voice interfaces should be subtle, context-sensitive, and non-disruptive.*

An important consideration emerging from our findings is that not all vishing interactions require the same response timing. Some attacks can cause immediate harm—such as requests for credit card numbers, one-time passwords, or account credentials—where the system must prioritize instant alerts to enable quick user action. Other interactions are part of longer trust-building sequences that precede the actual scam. In such cases, an immediate interruption may be unnecessary or even counterproductive. These scenarios call for *adaptive intervention strategies* that vary the timing and intensity of alerts according to the perceived threat immediacy. Post-call summaries and educational feedback may thus complement or, in lower-risk contexts, substitute in-call alerts, helping users reflect on suspicious patterns without disrupting legitimate communication.

Future conversational security systems should therefore explore dynamic feedback mechanisms that weigh both the urgency and context of potential threats. Adjusting the granularity and timing of notifications—ranging from discreet in-call cues to reflective post-call reports—can help ensure that interventions are both effective and user-appropriate, preserving trust and conversational flow while minimizing the likelihood of harm.

6.2 Explanatory and Educational Feedback

Beyond immediate alerts, our post-call reports served an explanatory and educational purpose, helping users understand why specific interactions were flagged. This feature addressed a key challenge in phishing prevention—the gradual decline of vigilance over time—by reinforcing awareness through personalized feedback. More broadly, our results point to the need for *transparent, instructive feedback mechanisms that transform detection into learning.* Integrating brief explanations, examples, or comparative summaries into post-interaction reports could help users develop a deeper understanding of social engineering tactics and build long-term resilience against manipulation.

6.3 Robustness and Inclusivity

Our evaluation showed consistent performance across accents, genders, and noise conditions, suggesting that well-tuned speech recognition and semantic analysis can support equitable protection. This emphasizes that *robustness is not only a technical concern but also an ethical one*: security systems that fail to account for linguistic or acoustic diversity risk excluding certain user groups. Designers should therefore evaluate voice-based defenses under diverse linguistic, cultural, and environmental conditions, ensuring that all users—regardless of background or setting—receive equal levels of protection. Extending such inclusivity to multilingual and offline contexts will be essential for broader accessibility.

6.4 Designing for a Changing Threat Landscape

Finally, our modular architecture—combining ASR, keyword detection, and LLM-based semantic analysis—proved adaptable to evolving social engineering tactics. This finding suggests a critical design implication: *security-oriented voice assistants should be built for continuous evolution.* Systems should support modular updates to keyword lists, LLM prompts, and model components, enabling designers to integrate new threat patterns without extensive redevelopment. Embedding versioning, feedback loops, and user reporting mechanisms can further support co-evolution between attackers' tactics and defense strategies.

At the same time, future work should look beyond reactive updates toward more fundamental forms of adaptability. Current approaches—such as keyword-based detection—are vulnerable to linguistic drift as scammers modify phrasing or invent new pretexts. Instead, future systems could focus on identifying higher-level conversational characteristics, such as *patterns of manipulation, pressure,*

or exaggerated helpfulness, that are less dependent on specific storylines or terminology. Detecting these behavioral and linguistic markers would make voice-based security systems more resistant to changing scam narratives, emphasizing psychological cues over surface features.

7 Conclusion

We presented the design, implementation, and evaluation of a real-time voice phishing detection system that integrates virtual voice assistants, automatic speech recognition, and large language models to safeguard users during phone conversations. By emphasizing content and context over caller identity or blacklists, the system overcomes limitations like spoofing and delayed updates. Evaluation showed robust performance across accents, genders, and noise, with 97.8% transcription accuracy via Whisper and 91.7% phishing classification via GPT-3.5 Turbo, achieving 8.7-second response times. Post-call reports enhance user awareness and resilience. The modular architecture enables compatibility with blacklists for hybrid defenses, cross-platform deployment, and extensions like multilingual support. Limitations include English-only operation, static keywords, and synthetic data reliance, warranting future work in adaptive learning, privacy-preserving inference, and real-world testing. This study highlights how NLP and voice technologies can address human-centric security challenges, particularly for vulnerable populations. It lays a foundation for secure voice interfaces, conversational threat detection, and interdisciplinary research in AI ethics and usability, positioning such systems as key to future cybersecurity infrastructures.

Acknowledgements This research is part of the *Voice of Wisdom* project and received funding from dtec.bw – Digitalization and Technology Research Center of the Bundeswehr. dtec.bw is funded by the European Union – NextGenerationEU.

References

1. Acosta, L.H., Reinhardt, D.: A survey on privacy issues and solutions for voice-controlled digital assistants. Pervasive and Mobile Computing **80**, 101523 (2022), https://doi.org/10.1016/j.pmcj.2021.101523
2. Armstrong, M.E., Jones, K.S., Namin, A.S.: How perceptions of caller honesty vary during vishing attacks that include highly sensitive or seemingly innocuous requests. Human Factors **65**(2), 275–287 (2023), https://doi.org/10.1177/00187208211012818
3. Astrakhantsev, A., Pedan, S.: Improving user security during a call. Radioelectronic and Computer Systems **2024**(2), 173–185 (2024), http://nti.khai.edu/ojs/index.php/reks/article/view/reks.2024.2.14
4. Babiker, A., Basta, T.: Deepfake voice implementation for scams: Information security risks with deepfake technology (2024), https://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-352483

5. Berens, B., Dimitrova, K., Mossano, M., Volkamer, M.: Phishing awareness and education–when to best remind. In: Workshop on Usable Security and Privacy (USEC) (2022), https://www.ndss-symposium.org/wp-content/uploads/usec2022_23075_paper.pdf
6. Berglund Molin, E.: The development of vishing fraud during the covid pandemic (2021), https://www.diva-portal.org/smash/get/diva2:1569367/FULLTEXT01.pdf
7. Bhattarai, K., Oh, I.Y., Sierra, J.M., Tang, J., Payne, P.R., Abrams, Z., Lai, A.M.: Leveraging gpt-4 for identifying cancer phenotypes in electronic health records: a performance comparison between gpt-4, gpt-3.5-turbo, flan-t5, llama-3-8b, and spacy's rule-based and machine learning-based methods. JAMIA open **7**(3), ooae060 (2024), https://doi.org/10.1093/jamiaopen/ooae060
8. Bolton, T., Dargahi, T., Belguith, S., Al-Rakhami, M.S., Sodhro, A.H.: On the security and privacy challenges of virtual assistants. Sensors **21**(7), 2312 (2021), https://doi.org/10.3390/s21072312
9. Chetioui, K., Bah, B., Alami, A.O., Bahnasse, A.: Overview of social engineering attacks on social networks. Procedia Computer Science **198**, 656–661 (2022), https://doi.org/10.1016/j.procs.2021.12.302
10. Cimino, G., Deufemia, V.: Towards enhanced human mitigation of vishing attacks: Leveraging large language models for real-time user guidance. In: CEUR Workshop Proceedings. vol. 3713 (2024), https://ceur-ws.org/Vol-3713/paper_6.pdf
11. Figueroa, C.: The pallone-thune" traced act": Expanding consumer protection in the fight against robocalls. Loy. Consumer L. Rev. **32**, 318 (2019), https://heinonline.org/HOL/LandingPage?handle=hein.journals/lyclr32&div=16&id=&page=
12. Graham, C., Roll, N.: Evaluating openai's whisper asr: Performance analysis across diverse accents and speaker traits. JASA Express Letters **4**(2) (2024), https://doi.org/10.1121/10.0024876
13. Hashmi, S.I., George, N., Saqib, E., Ali, F., Siddique, N., Kashif, S., Ali, S., Bajwa, N.U.H., Javed, M.: Training users to recognize persuasion techniques in vishing calls. In: Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems. pp. 1–8 (2023), https://doi.org/10.1145/3544549.3585823
14. Kalaharsha, P., Mehtre, B.M.: Detecting phishing sites–an overview. arXiv preprint arXiv:2103.12739 (2021), https://doi.org/10.48550/arXiv.2103.12739
15. Kim, D., Sehwan, O., Ban, Y., Park, J., Joo, K., Cho, H.: Ventinel: Automated detection of android vishing apps using optical character recognition. Future Internet **17**(1), 1–19 (2025), https://doi.org/10.3390/fi17010024
16. Kim, J., Kim, J., Wi, S., Kim, Y., Son, S.: Hearmeout: detecting voice phishing activities in android. In: Proceedings of the 20th Annual International Conference on Mobile Systems, Applications and Services. p. 422–435. MobiSys '22, Association for Computing Machinery, New York, NY, USA (2022). https://doi.org/10.1145/3498361.3538939, https://doi.org/10.1145/3498361.3538939
17. Lee, M., Park, E.: Real-time korean voice phishing detection based on machine learning approaches. Journal of Ambient Intelligence and Humanized Computing **14**(7), 8173–8184 (2023)
18. McEachern, J., Burger, E.: How to shut down robocallers: The stir/shaken protocol will stop scammers from exploiting a caller id loophole. IEEE Spectrum **56**(12), 46–52 (2019), https://doi.org/10.1109/MSPEC.2019.8913833

19. Pandit, S., Liu, J., Perdisci, R., Ahamad, M.: Fighting voice spam with a virtual assistant prototype. CoRR **abs/2008.03554** (2020), https://doi.org/10.48550/arXiv.2008.03554

20. Pandit, S., Perdisci, R., Ahamad, M., Gupta, P.: Towards measuring the effectiveness of telephony blacklists. In: NDSS (2018), http://dx.doi.org/10.14722/ndss.2018.23243

21. Park, S., Yoon, H., Kim, J., Kim, H., Lee, S.J.: "i know my data doesn't leave my phone, but still feel like being wiretapped": Understanding (mis)perceptions of on-device ai vishing detection apps. In: Proceedings of the Extended Abstracts of the CHI Conference on Human Factors in Computing Systems. CHI EA '25, Association for Computing Machinery, New York, NY, USA (2025). https://doi.org/10.1145/3706599.3719784, https://doi.org/10.1145/3706599.3719784

22. Schwab, J., Nussbaum, A., Sergeeva, A., Alt, F., Distler, V.: What makes phishing simulation campaigns (un) acceptable? a vignette experiment. In: Network and Distributed System Security Symposium, NDSS 2025 (2025), https://www.ndss-symposium.org/wp-content/uploads/usec25-10.pdf

23. Sharevski, F., Jachim, P.: Alexa in phishingland: Empirical assessment of susceptibility to phishing pretexting in voice assistant environments. In: 2021 IEEE Security and Privacy Workshops (SPW). pp. 207–213. IEEE (2021), https://doi.org/10.1109/SPW53761.2021.00034

24. Sharevski, F., Jachim, P.: "alexa, what'sa phishing email?": Training users to spot phishing emails using a voice assistant. EURASIP Journal on Information Security **2022**(1), 7 (2022), https://doi.org/10.1186/s13635-022-00133-w

25. Steinmetz, K.F., Pimentel, A., Goe, W.R.: Performing social engineering: A qualitative study of information security deceptions. Computers in Human Behavior **124**, 106930 (2021), https://doi.org/10.1016/j.chb.2021.106930

26. Teng, K.: Unmasking the villain: Exposing scammers' identities to defeat harmful calls. Brook. J. Corp. Fin. & Com. L. **14**, 367 (2019), https://heinonline.org/HOL/LandingPage?handle=hein.journals/broojcfc14&div=22&id=&page=

27. Terzopoulos, G., Satratzemi, M.: Voice assistants and smart speakers in everyday life and in education. Informatics in Education **19**(3), 473–490 (2020), https://www.ceeol.com/search/article-detail?id=896140

28. Toapanta, F., Rivadeneira, B., Tipantuña, C., Guamán, D.: Ai-driven vishing attacks: A practical approach. Engineering Proceedings **77**(1), 15 (2024), https://doi.org/10.3390/engproc2024077015

29. Wang, S., Delavar, M., Azad, M.A., Nabizadeh, F., Smith, S., Hao, F.: Spoofing against spoofing: Toward caller id verification in heterogeneous telecommunication systems. ACM transactions on privacy and security **27**(1), 1–25 (2023), https://doi.org/10.1145/3625546

30. Wang, Z., Zhu, H., Sun, L.: Social engineering in cybersecurity: Effect mechanisms, human vulnerabilities and attack methods. Ieee Access **9**, 11895–11910 (2021). https://doi.org/10.1109/ACCESS.2021.3051633

31. Yujian, L., Bo, L.: A normalized Levenshtein distance metric. IEEE Transactions on Pattern Analysis and Machine Intelligence **29**(6), 1091–1095 (2007). https://doi.org/10.1109/TPAMI.2007.1078

Disclosure of Interests

The authors have no competing interests to declare that are relevant to the content of this article.

Acknowledgment of AI Use

GPT-5 was used to rephrase text, refine writing style, and do grammar and spelling checks in order to improve the overall quality of the text. In addition, GPT-3.5 Turbo was used for semantic classification of keywords, and Whisper Automated Speech Recognition (ASR) was used to transcribe ongoing calls into segments.