An AI-Based Security Architecture for Fraud Detection in Cloud Call Centers for Low-Resource Languages: Arabic as a Use Case

Pinar Boluk * and Hana'a Maratouq

Posted Date: 25 March 2026

doi: 10.20944/preprints202603.1997.v1

Keywords: fraud detection; cloud telephony security; Arabic natural language processing (NLP); automatic speech recognition; large language models; threat model; low-resource languages

 doi:10.20944/preprints202603.1997.v1

Article

An AI-Based Security Architecture for Fraud Detection in Cloud Call Centers for Low-Resource Languages: Arabic as a Use Case

Pinar Boluk [1,*] and Hana'a Maratouq [2]

[1] Department of Artificial Intelligence and Data Engineering, Istanbul University, Istanbul 34134, Turkey
[2] Department of Computer Engineering, Bahçeşehir University, Istanbul 34353, Turkey
* Correspondence: pinar.boluk@istanbul.edu.tr

Abstract

Cloud-based telephony platforms face growing fraud risks including voice phishing (vishing), subscription abuse, and organizational impersonation, with detection being especially challenging in low-resource languages such as Arabic. This paper presents an Artificial Intelligence (AI)-based security architecture for fraud detection in Arabic cloud call centers, integrating onboarding verification, behavioral monitoring, domain-adapted Automatic Speech Recognition (ASR), semantic transcript search, and Large Language Model (LLM)-based entity verification. The domain-adapted Langa ASR model achieves a Word Error Rate (WER) of 41.0% and Character Error Rate (CER) of 18.2%, outperforming all evaluated commercial baselines. LLM-based entity extraction with multi-call consensus achieves 97.3% company-name accuracy (Generative Pre-trained Transformer 4, GPT-4) and 92.0% in the cost-effective deployed configuration (GPT-3.5 with log-probability filtering). Evaluated on production data from a Middle East and North Africa (MENA)-region provider spanning more than 1,000 accounts, the pipeline flagged 47 accounts of which 41 were confirmed fraudulent (precision 87.2%, 95% confidence interval (CI): 74.3%–95.2%; estimated recall 51%–82%), demonstrating the viability of a unified, threat-model-driven architecture for low-resource telephony fraud detection.

Keywords: fraud detection; cloud telephony security; Arabic natural language processing (NLP); automatic speech recognition; large language models; threat model; low-resource languages

1. Introduction

The rapid expansion of cloud-based communication services is reshaping the digital infrastructure of the global economy. Small, medium, and large enterprises increasingly rely on cloud call centers for customer interactions, telemarketing campaigns, and distributed support services [1]. Internet-based remote call centers are proliferating as businesses seek to reduce infrastructure costs and enhance operational flexibility [2].

The same properties that make cloud telephony attractive—scalability, geographic distribution, and ease of provisioning—also lower the barrier for abuse. Fraudulent actors exploit these platforms through subscription fraud using forged or stolen credentials [3], voice phishing (vishing) campaigns in which agents impersonate legitimate organizations to extract sensitive information [4], and refund or free-trial exploitation. Beyond direct financial loss, such fraud exposes providers to regulatory penalties and erodes end-user trust in digital services [5].

Despite the severity of the threat, existing cloud telephony fraud detection systems share three structural gaps. First, Call Detail Record (CDR)-based and rule-based approaches [6,7] are blind to voice-driven fraud encoded in call content rather than call metadata [8,9,14]. Second, content-aware Natural Language Processing (NLP) systems [12,13] are evaluated on controlled single-language datasets and do not generalize to morphologically complex Arabic telephony. Third, prior systems lack an explicit security threat model or production validation. These gaps are compounded by the

 doi:10.20944/preprints202603.1997.v1

inherent difficulty of processing Arabic speech at scale. Conversational fraud detection in Arabic faces linguistic obstacles absent in well-resourced languages: a wide dialect continuum (Egyptian, Levantine, Gulf, Maghrebi), pervasive code-switching with Modern Standard Arabic (MSA), and rich morphology that degrades ASR quality—the foundational step for all downstream analytics. Section 2 provides a detailed comparative analysis of both the fraud detection and Arabic NLP literature.

This paper addresses the gap between isolated research contributions and operational security deployment by presenting a security-by-design, end-to-end fraud detection architecture for Arabic cloud call centers. Specifically, the contributions of this work are threefold:

1. **Hybrid transcript-based fraud detection for Arabic:** A two-phase proposed pipeline detects fraud-related conversational intent in Arabic call transcripts despite dialectal and morphological variation. The pipeline combines rule-based keyword filtering with neural embedding-based semantic retrieval over MSA-normalized call summaries, providing complementary precision and recall coverage.
2. **Systematic LLM ablation for behavioral and entity verification:** In this work, GPT-3.5, GPT-4, and the Arabic-centric Jais model are evaluated for agent and company-name extraction across prompt structure (few-shot vs. zero-shot), segment selection, multi-call consensus, and log-probability confidence filtering. Results quantify the accuracy–cost trade-off and identify a cost-effective operational configuration.
3. **End-to-end production validation:** The proposed pipeline is evaluated on production data from a MENA-region provider (1,024 accounts, ∼38,000 calls). Reported metrics include precision with Wilson score confidence intervals, conservative recall bounds accounting for the metadata escalation funnel, a confusion matrix, and per-layer adversarial resistance analysis across all four identified attack surfaces (Sections 3 and 6).

The remainder of this paper is organized as follows. Section 2 reviews related work critically. Section 3 presents the threat model and security assumptions. Section 4 describes the proposed architecture. Section 5 presents experimental results. Section 6 provides the security analysis. Section 7 discusses implications and limitations. Section 8 concludes.

2. Related Work

Fraud detection research spans three bodies of literature relevant to this work: financial fraud systems, telecom CDR analytics, and conversational NLP approaches. Each addresses a partial aspect of the problem; none covers the full combination of conversational intelligence, Arabic language support, adversarial threat modeling, and production validation that cloud telephony fraud demands.

Financial fraud detection has been extensively studied, with rule-based systems [33], ensemble learning, and graph neural networks applied to credit card, money laundering, and loan fraud detection [34–36]. Real-time big data pipelines integrating Spark, Kafka, and Isolation Forest demonstrate practical scalability in transactional systems [37]. However, these approaches are *architecturally incompatible* with cloud telephony fraud. Transactional fraud leaves structured digital traces (amounts, merchants, timestamps) amenable to statistical modeling; telephony fraud is primarily a social-engineering phenomenon whose evidence is embedded in natural language utterances. Applying financial fraud detectors to call-center fraud detection therefore addresses the wrong threat model.

Within telecommunications, CDR analytics remains the dominant fraud detection approach [6,7, 42]. FrauDetector [10] applies a weighted Hyperlink-Induced Topic Search (HITS) algorithm to call graphs to propagate phone trust values—effective for Subscriber Identity Module (SIM)-box fraud, toll bypass, and infrastructure-level abuse, but fundamentally incapable of detecting impersonation: a fraudulent agent who makes calls at normal volume, frequency, and routing pattern is completely invisible to any CDR-based system. Latent Dirichlet Allocation (LDA)-based user profiling [11] and Support Vector Machine (SVM)/Artificial Neural Network approaches applied to CDR features [38] share this limitation. The critical insight missing from this body of work is that *cloud telephony fraud is intentionally designed to mimic legitimate calling behavior at the metadata level*—subscription fraud

 doi:10.20944/preprints202603.1997.v1

accounts pass CDR screening by design, making conversational intelligence architecturally necessary, not optional.

Conversational and NLP-based approaches address this gap to a degree. NLP-based methods detect semantic fraud cues such as urgency language, impersonation claims, and financial solicitation [12]. Graph-based LLM approaches identify fraudulent entities within interaction networks [26]. Multimodal frameworks integrating acoustic and textual signals improve robustness against voice phishing [13], and ASR-to-text NLP pipelines demonstrate vishing detection in Korean, confirming that low-resource settings require domain-focused data collection and adapted models [39]. Nevertheless, all existing transcript-based fraud detection systems are evaluated on controlled, single-language datasets and share three critical gaps: none addresses a morphologically complex low-resource language at production scale, none incorporates explicit adversarial threat modeling, and none reports production deployment results with statistical validation.

On the Arabic NLP side, well-documented ASR challenges include dialectal diversity, morphological richness, code-switching behavior, and telephony channel noise [15,16]. Recent studies confirm that fine-tuning multilingual models on call-domain speech data significantly improves transcription reliability [17,18,45], and that improved transcription propagates directly to better downstream NLP accuracy [19]. Semantic embedding and transformer-based retrieval enable cross-dialectal similarity search at scale [22,25]. Yet no prior work integrates these capabilities into a deployed security architecture with explicit threat modeling.

Table 1 summarizes how the present work extends the state of the art across five dimensions critical to operational security deployment; ✓ indicates fully addressed, ○ partially addressed, and × not addressed. As shown, Table 1 summarizes coverage across five dimensions.

Table 1. Comparison of fraud detection approaches across five dimensions critical to secure cloud telephony deployment.

Approach	Conversational Intelligence	Arabic Language	Threat Model	Deployment Validation	Eval. Rigor
CDR analytics [6,7]	×	○	×	✓	○
FrauDetector [10]	×	×	×	✓	○
NLP transcripts [12]	✓	×	×	×	×
Graph LLM [26]	✓	×	×	×	○
This work	✓	✓	✓	✓	✓

3. Threat Model and Security Assumptions

Security-oriented fraud detection systems must be designed against an explicit adversary model. We define the threat model for cloud telephony fraud detection following standard security analysis conventions.

3.1. Adversary Goals

We consider adversaries with two primary operational goals:

Goal G1 — Financial exploitation: The adversary aims to provision and operate cloud telephony services (Direct Inward Dialing (DID) numbers, trunk capacity) without legitimate payment, exploiting stolen payment instruments, free trials, or refund mechanisms.

Goal G2 — Vishing and impersonation: The adversary aims to conduct outbound calling campaigns in which agents falsely represent themselves as employees of legitimate organizations to solicit sensitive information, financial transactions, or behavioral compliance from call recipients.

3.2. Adversary Capabilities

What the adversary controls:

- *Account registration data*: The adversary can fabricate or steal business credentials, email addresses, company registrations, and payment instruments used during onboarding.

 doi:10.20944/preprints202603.1997.v1

- *Agent speech content*: The adversary fully controls what agents say during calls, including the organizational identity they claim, the scripts they use, and the urgency or authority cues they deploy.
- *Call metadata*: The adversary can control call volume, call timing, and callee selection to mimic legitimate calling patterns, partially evading CDR-based detectors.
- *Multiple accounts*: The adversary can operate multiple subscriber accounts, potentially using shared or rotating payment instruments, to distribute calling activity and reduce per-account anomaly scores.

3.3. Attack Surface Definition

We identify four primary attack surfaces in the proposed architecture:

AS1 — Onboarding evasion: The adversary provides plausible but fabricated documentation to pass onboarding verification. Countermeasure: multi-source cross-field validation and behavioral urgency signals (see Section 4.2).

AS2 — Metadata camouflage: The adversary patterns calling behavior to resemble legitimate accounts, suppressing CDR-level anomaly signals. Countermeasure: the architecture's cascade design means metadata evasion alone is insufficient; accounts must also pass conversational verification.

AS3 — ASR evasion: The adversary attempts to degrade ASR accuracy by introducing noise, accents, or obfuscated speech patterns to prevent keyword and entity extraction. Countermeasure: embedding-based semantic search is robust to ASR errors (operates on summary-level semantics, not exact transcript tokens); the domain-adapted Langa model is explicitly trained on telephony-condition speech.

AS4 — LLM prompt injection: The adversary attempts to manipulate LLM extraction by embedding adversarial text in call content (e.g., an agent says phrases designed to confuse the LLM prompt or inject false entity names) [40]. Countermeasure: LLM prompts operate on short, structured segments; multi-call consensus requiring agreement across temperature-varied queries substantially reduces susceptibility to single-injection attacks; logprob filtering rejects low-confidence extractions.

4. Architecture and Methodology

4.1. System Overview and Design Principles

The architecture employs a two-path cascade: Path 1 applies low-cost metadata screening for early risk estimation and anomaly detection; Path 2 applies call-content analysis to detect fraud embedded in agent speech. The cascade design serves both computational efficiency—expensive speech and language processing is reserved for accounts that exceed a configurable escalation threshold—and security depth, since adversaries must evade multiple independent mechanisms simultaneously.

Formal definition. Given account registration data $\mathcal{A}$, a metadata stream $\mathcal{M}$ (billing events, usage logs, configuration changes), and a set of call recordings $\mathcal{C}$, the system produces a binary risk flag $y \in \{0, 1\}$ and an evidence report $\mathcal{R}$. The report-oriented design reflects operational practice: rather than taking unilateral account-suspension actions, the system structures evidence to support human review and auditable enforcement decisions.

Figure 1 illustrates the complete pipeline. Path 1 (metadata screening) gates accounts through an escalation threshold before the costlier Path 2 (speech and content analysis); attack surfaces AS1–AS4 are annotated at the module boundaries where each is exploitable. The decision layer aggregates all evidence into report $\mathcal{R}$ and routes flagged accounts to human validation rather than taking unilateral enforcement action.

 doi:10.20944/preprints202603.1997.v1

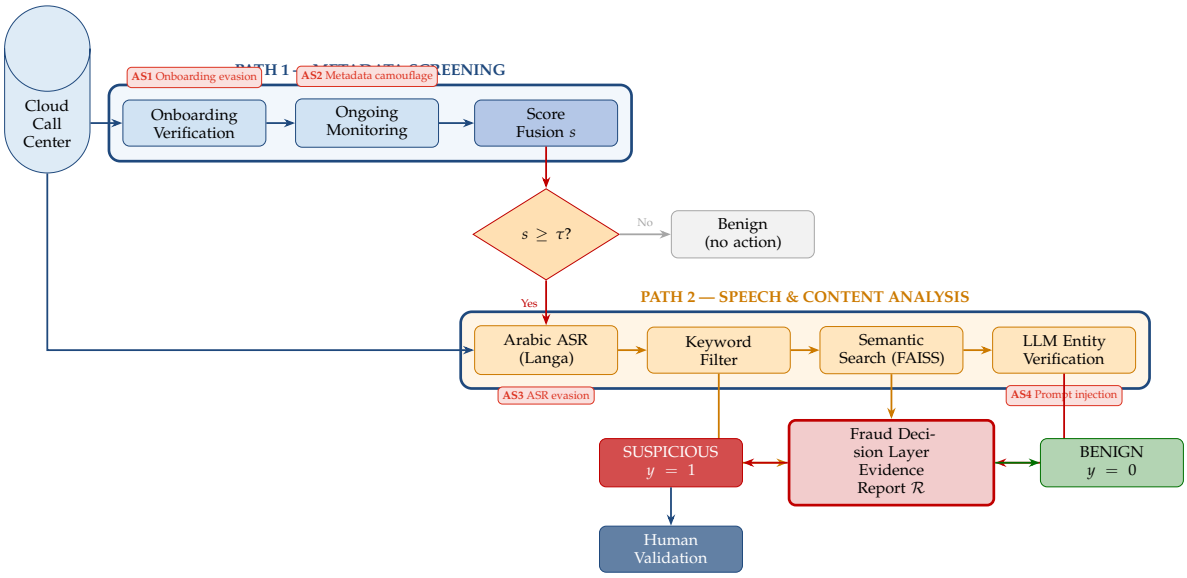

Figure 1. End-to-end fraud detection security architecture with attack surfaces AS1–AS4 annotated at each vulnerable module boundary.

4.2. Metadata-Based Detection Path

Onboarding Verification (Countermeasure: AS1).

Onboarding verification aims to reduce exposure before fraud campaigns begin. A risk score s_{meta} is computed from five signal groups: document completeness, cross-field consistency, email and domain integrity, website legitimacy, and behavioral urgency at registration. Algorithm 1 formalizes this computation; the output is a normalized score $s_{\text{meta}} \in [0, 1]$ and a set of human-readable flags added to $\mathcal{R}$.

Ongoing Monitoring (Countermeasure: AS2).

Fraud frequently occurs after a seemingly valid onboarding, particularly when adversaries exploit free trials, stolen payment methods, or operational loopholes. The monitoring component tracks four anomaly signals: shared or previously flagged payment instruments, callee overlap across accounts indicating coordinated control, operational misuse patterns (e.g., one account driving calls for multiple businesses), and DID cross-checks against blacklist services such as Truecaller. These signals produce a time-varying anomaly score s_{mon}.

Escalation Logic.

The metadata path outputs a fused score:

$$s = \text{FUSESCORES}(s_{\text{meta}}, s_{\text{mon}}), \tag{1}$$

where FUSESCORES$(\cdot)$ is implemented as a weighted sum $s = \alpha \cdot s_{\text{meta}} + (1-\alpha) \cdot s_{\text{mon}}$. The weight α and the escalation threshold τ_{escalate} are calibrated experimentally on a labeled subset of historical accounts by maximizing the F1 score of the escalation decision, yielding the observed 15–20% escalation rate in the production deployment. Accounts with $s \geq \tau_{\text{escalate}}$ are passed to the call-content path.

4.3. Call-Content-Based Detection Path

Automatic Speech Recognition.

Five commercial and open-source ASR systems were evaluated:

- **Meta M4T V1:** A massively multilingual model supporting 200 languages with encoder-decoder architecture and mean-pooling representations [29].
- **Whisper Large V1:** Trained on 680,000 hours of multilingual audio [30], using encoder-decoder Transformer with log-Mel spectrogram input.

 doi:10.20944/preprints202603.1997.v1

- **Chirp:** Google's ASR technology optimized for challenging noisy environments [31].
- **Google Cloud Speech-to-Text Application Programming Interface (API):** Wide language support including Arabic dialects with real-time transcription.
- **Microsoft Azure Speech-to-Text API:** Customizable Arabic speech recognition across audio environments.

Additionally, **Langa** [32], a Whisper-backbone model fine-tuned on the target telephony domain using progressive domain adaptation [20,21], was evaluated as the candidate deployment model.

ASR Benchmark Dataset.

The benchmark dataset was developed through a meticulous annotation pipeline involving 41 annotators and 13 reviewers, producing 132 hours of high-quality, multi-dialect Arabic telephony speech [19]. Calls were sourced from a cloud telephony provider operating across the Arab region with full user consent; all sensitive business information was removed prior to annotation. The dataset spans **six Arabic dialect groups** (Egyptian, Levantine, Gulf, Iraqi, Maghrebi, and MSA-dominant) representing **13 countries**, and covers diverse application domains including education, entertainment, and e-commerce. Audio is standardized at 16 kHz and encompasses a wide dynamic range of channel conditions—from clean studio-quality recordings to heavily degraded telephony noise—ensuring realistic evaluation of ASR robustness in production deployments. Annotation quality was validated through inter-annotator agreement checks conducted by the reviewer panel.

Evaluation Methodology.

ASR quality is assessed using WER and CER. Blind human comparisons are conducted in which annotators vote for the better transcript without knowledge of model identity.

Operational Throughput and Privacy.

The Langa model was profiled on the production deployment environment. Processing a 1.5-minute call (the mean call duration) requires approximately 3–5 seconds on a single NVIDIA T4 Graphics Processing Unit (GPU), yielding a real-time factor (RTF) of $\approx$ 0.03–0.06. This supports processing of up to 80,000 calls/day with 2–3 parallel GPU workers—consistent with typical medium-sized cloud call center volumes—via asynchronous batch processing with no latency constraint on the fraud decision. All data are processed under a personally identifiable information (PII)-minimization principle: only features required for detection are retained, manual review is restricted to escalated accounts, and call recordings were sourced with user consent [19]. This privacy-by-design approach aligns with General Data Protection Regulation (GDPR) and Personal Data Protection Law (PDPL) requirements applicable in the MENA region [44].

Keyword Filtering for High-Risk Lexical Cues.

A lightweight exact-match keyword search serves as a fast triage stage. A fraud-related Arabic keyword list is employed, corresponding to terms translating to: *scam, fraud, password, card CVC/number, OTP/verification code, complaint, communications commission*, and *theft*. In Arabic: [nasb, ihtial, murur, ramz bitaqa, ramz tahaqquq, shakwa, hayat al-ittisal, sariqa].

For each account, normalized keyword hit counts are computed over its call corpus, producing a lexical-risk vector $\mathbf{k}(\mathcal{A})$. Keyword counts are not treated as definitive proof of fraud; they contribute to $\mathcal{R}$ and prioritize accounts for deeper analysis. Legitimate business vocabulary overlap motivates combination with semantic retrieval.

Semantic Search over Standardized Call Summaries.

Semantic retrieval is implemented to address the brittleness of exact matching in Arabic (Counter-measure: AS3 at the semantic level). Figure 2 shows the two-phase retrieval pipeline. In the **offline phase** (steps 1–5), call recordings are transcribed by the Langa ASR model, summarized in Modern Standard Arabic to normalise dialectal variation, encoded into dense vectors by AraBERT, and stored

 doi:10.20944/preprints202603.1997.v1

in a Facebook AI Similarity Search (FAISS) Inverted File (IVF) index; this build step requires approximately 15 minutes for 100k summaries and is performed once. In the **online phase** (steps A–D), an analyst query is encoded with the same AraBERT weights (shared encoder) and matched against the index via Approximate Nearest Neighbor (ANN) cosine-similarity search, returning the top-K most relevant call summaries in under 5 ms on Central Processing Unit (CPU) (FAISS IVF, nprobe = 64).

Retrieval Targets.

Summaries are searched to identify: (i) complaints about scams, spam, or fraud; (ii) discussion deviating from known account use-cases; (iii) personal conversations inconsistent with business operations; and (iv) solicitation of financial or personal data.

Embedding and Indexing.

Summaries and queries are encoded using Arabic-capable sentence-transformer models: `bert-base-arabic` and `bert-base-arabertv2`[1] [22,23]. Similarity is computed via cosine distance. ANN search uses the FAISS library [24] with an inverted-file index. The semantic retrieval layer outputs top-K summaries with similarity scores.

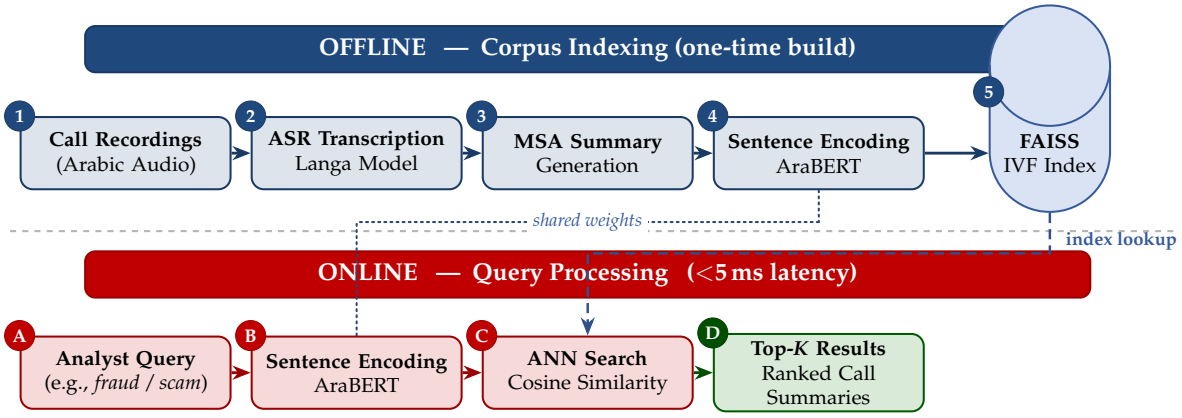

Figure 2. Two-phase semantic retrieval pipeline: offline corpus indexing (steps 1–5) and online query processing (steps A–D) using shared AraBERT embeddings and FAISS ANN search.

Algorithm 1 formalizes the onboarding risk scoring procedure. Each of the five stages contributes a calibrated penalty $w_i \in (0, 1]$ to a cumulative score s_{meta}, which is clipped to $[0, 1]$. The resulting flag set F is appended to the evidence report $\mathcal{R}$ for downstream human review. The per-stage weights (Table 2) are calibrated on a labeled subset of historical accounts by maximizing escalation F1 on a held-out 20% split.

Behavioral and Entity-Based Verification via LLMs.

Many cloud call-center fraud campaigns involve misrepresentation: agents claim affiliation with a legitimate organization or provide inconsistent identity details. The behavioral analysis stage verifies whether agent self-introductions match registered account information.

Datasets.

Two manually labeled datasets are used for evaluation. *Dataset-165* consists of 165 call transcripts sourced from a MENA-region cloud telephony provider; each transcript was manually annotated with the agent name and organization name as stated during the agent's self-introduction. Calls originate from diverse accounts and are primarily in Arabic, with occasional code-switching. *Dataset-50* is an independently labeled set of 50 calls constructed *after* prompt optimization to assess generalization

[1] https://huggingface.co/asafaya/bert-base-arabic, https://huggingface.co/aubmindlab/bert-base-arabertv2

 doi:10.20944/preprints202603.1997.v1

and detect potential prompt bias; it was not used during prompt development. Both datasets consist of real production calls, not synthetic or crowd-sourced transcripts.

Algorithm 1 Onboarding Risk Score Computation

Require: Account registration data $\mathcal{A}$ (documents, email, website, cross-fields, behavioral signals)
Ensure: Onboarding risk score $s_{\text{meta}} \in [0, 1]$, flag set F

$F \leftarrow \emptyset$; $s_{\text{meta}} \leftarrow 0$
// Stage 1: Document completeness check
if $\mathcal{A}$.documents incomplete **or** any required field $\in \emptyset$ **then**
    $s_{\text{meta}} \mathrel{+}= w_1$; $F \leftarrow F \cup \{\text{MISSING\_DOCS}\}$
end if
// Stage 2: Email and domain integrity
if $\mathcal{A}$.email matches free-provider list **or** $\mathcal{A}$.email_domain $\neq$ $\mathcal{A}$.company_domain **then**
    $s_{\text{meta}} \mathrel{+}= w_2$; $F \leftarrow F \cup \{\text{EMAIL\_RISK}\}$
end if
// Stage 3: Website legitimacy
if $\mathcal{A}$.website unreachable **or** contains placeholder text (e.g., "Lorem ipsum") **then**
    $s_{\text{meta}} \mathrel{+}= w_3$; $F \leftarrow F \cup \{\text{FAKE\_WEB}\}$
end if
// Stage 4: Cross-field consistency
if $\mathcal{A}$.account_name $\not\approx$ $\mathcal{A}$.company_name **or** billing mismatch **then**
    $s_{\text{meta}} \mathrel{+}= w_4$; $F \leftarrow F \cup \{\text{INFO\_MISMATCH}\}$
end if
// Stage 5: Behavioral urgency signals at registration
if $\mathcal{A}$.urgency_score $> \tau_{\text{urgency}}$=0.6 **or** non-standard payment method requested **then** ▷ urgency_score: normalised count of urgency-signalling keywords in registration notes
    $s_{\text{meta}} \mathrel{+}= w_5$; $F \leftarrow F \cup \{\text{URGENCY}\}$
end if
$s_{\text{meta}} \leftarrow \min(s_{\text{meta}}, 1.0)$ ▷ Weights $w_i \in (0, 1]$ are calibrated on historical labeled accounts
return s_{meta}, F

Table 2. Onboarding risk score weights w_i (calibrated on labeled historical accounts by maximizing escalation F1 on a held-out 20% split).

Stage	Signal	Weight w_i
1	Missing documents	0.35
2	Email/domain risk	0.20
3	Fake or unreachable website	0.20
4	Cross-field inconsistency	0.15
5	Registration urgency	0.10

Task-Specific Named Entity Recognition (NER) Models.

Arabic BERT-based NER models demonstrate strong performance on standard benchmarks (F1: 80–94% across MSA, Classical, and dialectal variants) [41], but performance degrades significantly on noisy conversational ASR transcripts. The best-performing Hugging Face model was fine-tuned to recognize names from the initial 200 characters of calls, where agent introductions typically occur. Extracted entities are matched to reference names using the FuzzyWuzzy Levenshtein-distance library; manual verification is conducted on all matches with similarity $\geq$ 20%. Evaluation uses *extraction accuracy*—the fraction of calls where the extracted name matches the ground-truth label—reported separately for agent names and company names. Precision, recall, and F1-score are also tracked: precision measures whether extracted entities are correct, recall measures coverage of labeled entities, and F1 provides a balanced summary.

 doi:10.20944/preprints202603.1997.v1

LLM-Based Extraction (Countermeasure: AS4).

Large language models are prompted to extract agent and company names. Experiments are conducted with GPT-3.5, GPT-4 [27], and Jais [28]—a 13-billion parameter model pre-trained on Arabic and English. The following verification strategies are evaluated: (i) *prompt-structure*: few-shot versus zero-shot; (ii) *segment selection*: agent-only versus agent-plus-customer; (iii) *multi-call consensus*: multiple LLM calls with varied temperature, accepting only cases where answers agree; (iv) *logprob filtering*: log-probability analysis with a 95% linear probability threshold to prioritize high-confidence extractions.

Reproducibility and Model Versioning.

All GPT experiments were conducted via the OpenAI API using the `gpt-3.5-turbo` and `gpt-4` model endpoints, accessed during the period Q4 2023–Q1 2024. Temperature is set to 0.0 for the primary extraction call (Prompt A: deterministic, few-shot) and 0.9 for the consensus divergence call (Prompt B: stochastic, zero-shot); this deliberate asymmetry maximises the chance of detecting unreliable extractions through disagreement. The maximum token budget is 150 output tokens per call, sufficient for name extraction tasks. Jais [28] was accessed via its publicly hosted API (13B-parameter checkpoint); inference cost is dominated by hosting rather than per-call API pricing. String matching uses FuzzyWuzzy v0.18 with Levenshtein distance; the similarity threshold of 40 was fixed *before* the main experiments based on a manual review of 30 development examples and was not adjusted thereafter. Because OpenAI API model weights are updated periodically, exact numerical reproduction requires access to a frozen snapshot; we report all hyperparameters and prompts in sufficient detail to enable approximate replication with any instruction-following LLM of comparable capability.

Entity Matching and Coverage.

Extracted entities are verified against registered account metadata using FuzzyWuzzy (Levenshtein distance) with a threshold of 40. This threshold accommodates LLM inconsistencies such as partial names or morphological variants common in Arabic (e.g., elision of the definite article *al-*, dialectal vowel shifts). In addition to *extraction accuracy* (correctness on matched cases), we report *coverage*—the proportion of calls for which the model returns a high-confidence answer rather than abstaining. The interplay between accuracy and coverage defines the operational cost–precision trade-off: high-confidence filtering raises accuracy at the cost of reduced coverage, with unmatched cases escalated to human review.

4.4. Decision Layer, Evidence Fusion, and Human Validation

Evidence Report Construction.

The system accumulates: (i) keyword hits with contextual excerpts; (ii) retrieved summaries with similarity scores; (iii) extracted entities with confidence proxies; and (iv) verification scores comparing extracted organizations to the registered name. The resulting report $\mathcal{R}$ supports operational triage with account-level synopsis, call-level highlights, and entity consistency comparisons.

Decision Rule.

$$y = \mathbb{I}\big[\text{DECISIONRULE}(\mathcal{R}) = \text{SUSPICIOUS}\big], \tag{2}$$

where conservative thresholds minimize false positives. $y = 1$ triggers an escalation rather than automatic enforcement.

Human-in-the-Loop Validation Protocol.

When an account is flagged ($y = 1$), a fraud analyst reviews the top-evidence calls and structured report $\mathcal{R}$. The analyst confirms or rejects the suspicious classification. This protocol provides: (a) a final recall correction layer that can recover accounts missed by automated stages; (b) adversarial

 doi:10.20944/preprints202603.1997.v1

robustness against subtle evasion attempts that automated systems might accept; and (c) an auditable decision record for regulatory compliance. Algorithm 3 summarizes the complete pipeline.

Algorithm 2 LLM Multi-Call Consensus Entity Verification

Require: Transcript t, registered org $\mathcal{A}$.org, number of calls N=3, logprob threshold θ=0.95, fuzzy threshold ϕ=40
Ensure: Entity label $\ell \in \{\text{MATCH}, \text{MISMATCH}, \text{UNKNOWN}\}$
responses $\leftarrow$ []
for $i = 1$ **to** N **do**
  temp $\leftarrow$ 0.0 if $i = 1$ else 0.9 ▷ Call 1: deterministic few-shot (Prompt A); Calls 2–3: stochastic zero-shot (Prompt B)
  prompt $\leftarrow$ PROMPT_A if $i = 1$ else PROMPT_B
  $(e_i, \pi_i) \leftarrow$ LLM_EXTRACT$(t[0{:}200\text{chars}], \text{prompt}, \text{temp})$ ▷ Restrict to agent-intro segment; π_i is mean token log-probability
  responses.append$((e_i, \pi_i))$
end for
// Consensus check: all N extractions must agree
if $|\{e_i : (e_i, \pi_i) \in \text{responses}\}| = 1$ **then** ▷ All calls returned identical entity string
  $\bar{\pi} \leftarrow \frac{1}{N} \sum_{i=1}^{N} \exp(\pi_i)$ ▷ Convert mean log-prob to linear probability
  if $\bar{\pi} \geq \theta$ **then** ▷ High-confidence consensus
    $e^* \leftarrow e_1$
    $v \leftarrow$ FUZZYWUZZY$(e^*, \mathcal{A}.\text{org})$ ▷ Levenshtein-based partial ratio; accommodates Arabic morphological variants
    if $v \geq \phi$ **then**
      return MATCH
    else
      return MISMATCH ▷ Agent claimed organization differs from registered name: impersonation signal
    end if
  end if
end if
return UNKNOWN ▷ No consensus or low confidence; case escalated to human review

Algorithm 2 describes the LLM-based multi-call consensus procedure used to verify whether an agent's declared organizational identity matches the registered account name. The algorithm issues N=3 independent extraction calls to the LLM for the same transcript segment. The first call uses a deterministic temperature (temp $= 0.0$) with a few-shot prompt (Prompt A), which maximises extraction consistency and serves as the primary answer. The subsequent two calls use a high-temperature stochastic setting (temp $= 0.9$) with a zero-shot prompt (Prompt B); these calls act as independent divergence probes. Restricting input to the first 200 characters of the transcript targets the agent self-introduction segment, where organizational claims are made most explicitly. Consensus is declared only if all N calls return the identical entity string. This deliberate asymmetry between a deterministic anchor call and stochastic verification calls substantially reduces susceptibility to hallucination: a fabricated or inconsistent entity will rarely survive agreement across all three calls. The mean linear log-probability $\bar{\pi}$ provides an additional confidence gate; extractions with $\bar{\pi} < \theta$=0.95 are rejected as uncertain and escalated to human review (UNKNOWN). For accepted extractions, FuzzyWuzzy Levenshtein matching with threshold ϕ=40 accommodates common Arabic morphological variants such as elision of the definite article *al-* and dialectal vowel shifts. A MISMATCH label—where the extracted organization name does not match the registered account name—constitutes a strong impersonation signal fed into the evidence report $\mathcal{R}$.

 doi:10.20944/preprints202603.1997.v1

Algorithm 3 End-to-End Fraud Detection Pipeline

Require: Account $\mathcal{A}$, metadata stream $\mathcal{M}$, call audio set $\mathcal{C}$, thresholds $\tau_{\text{escalate}}, \tau_{\text{flag}}$
Ensure: Fraud flag $y \in \{0,1\}$, evidence report $\mathcal{R}$

$\mathcal{R} \leftarrow \emptyset$ ▷ Initialize empty evidence accumulator
// Path 1: Metadata Screening
$s_{\text{meta}}, F_{\text{onboard}} \leftarrow \text{ONBOARDINGVERIFY}(\mathcal{A})$ ▷ Algorithm 1
$s_{\text{mon}} \leftarrow \text{MONITORANOMALIES}(\mathcal{M})$ ▷ Three CDR signals: payment-account sharing, callee-set overlap, burst call-rate; see §4 AS2
$s \leftarrow \alpha \cdot s_{\text{meta}} + (1-\alpha) \cdot s_{\text{mon}}$ ▷ Weighted fusion; α=0.6 (calibrated on labeled history; Path 1 given higher weight as lower-cost signal)
$\mathcal{R}.\text{add}(s_{\text{meta}}, s_{\text{mon}}, F_{\text{onboard}})$
if $s < \tau_{\text{escalate}}$ **then**
    return $(y{=}0, \mathcal{R})$ ▷ Account passes metadata screening; no content analysis
end if
// Path 2: Call-Content Analysis (escalated accounts only)
for each call $c \in \mathcal{C}$ **do**
    $t \leftarrow \text{LANGA\_ASR}(c)$ ▷ Domain-adapted Whisper; RTF $\approx$ 0.03–0.06 on NVIDIA T4
    $\mathbf{k} \leftarrow \text{KEYWORDFILTER}(t, \text{ArabicKeywordList})$ ▷ 8 fraud-related terms; normalized hit count
    $q \leftarrow \text{SUMMARIZE}(t)$ ▷ MSA-normalized call summary for cross-dialectal retrieval
    $\mathbf{v}_q \leftarrow \text{ARABERT\_ENCODE}(q)$
    $\mathcal{N} \leftarrow \text{FAISS\_ANN}(\mathbf{v}_q, K{=}10)$ ▷ Top-K similar summaries; IVF index, nprobe=64, <5 ms
    $\ell \leftarrow \text{CONSENSUS\_VERIFY}(t, \mathcal{A}.\text{org})$ ▷ Algorithm 2; N=3, θ=0.95, ϕ=40
    $\mathcal{R}.\text{add}(\mathbf{k}, \mathcal{N}, \ell)$
end for
// Decision and Human Validation
$r_{\text{fraud}} \leftarrow |\{c : \ell(c) = \text{MISMATCH}\}| \,/\, |\mathcal{C}|$
$y \leftarrow \mathbb{I}[r_{\text{fraud}} \geq \tau_{\text{flag}}$ **or** $\bar{s}_{\text{sem}}(\mathcal{R}) \geq \tau_{\text{sem}}]$ ▷ τ_{flag}=0.4; τ_{sem}=0.72 (cosine similarity mean over top-K neighbors, tuned on validation set)
if $y = 1$ **then**
    $\text{HUMANVALIDATE}(\mathcal{A}, \mathcal{R})$ ▷ Analyst reviews top evidence calls and structured report
end if
return $(y, \mathcal{R})$

Algorithm 3 integrates Path 1 (metadata screening), Path 2 (call-content analysis), and human validation into a single executable procedure. Key implementation parameters are: escalation weight α (calibrated on labeled history), τ_{escalate} (threshold yielding 15–20% escalation rate), K=10 ANN neighbors, and τ_{flag} (maximizing precision on validation set). Flagged accounts are routed to human review rather than automatic enforcement, preserving an auditable decision record.

5. Results

5.1. ASR Performance

Table 3 presents the comparative WER and CER across all evaluated ASR systems. The domain-adapted Langa model achieves the lowest WER (41.0%) and CER (18.2%), outperforming all commercial baselines. Compared to the strongest commercial baseline (Chirp), Langa reduces WER by 7.9 percentage points and CER by 4.2 percentage points. The weakest commercial model (Whisper Large V1) lags Langa by 42.8 percentage points in WER, underscoring the substantial advantage of telephony-domain fine-tuning.

 doi:10.20944/preprints202603.1997.v1

Table 3. Comparative ASR performance (WER and CER).

Model	WER (%)	CER (%)
Langa (domain-adapted)	**41.0**	**18.2**
Chirp	48.9	22.4
Meta M4T V1	67.8	34.3
Google API	67.1	40.6
Azure API	71.9	39.0
Whisper Large V1	83.8	52.3

Blind listening experiments confirm these gains perceptually: annotators preferred Langa over Meta M4T in 99.1% of pairwise comparisons, over Google in 97.3%, and over Azure in 92.4% (Table 4). Based on these results, Langa was selected as the ASR model for the operational pipeline.

Table 4. Blind human listening comparisons: Langa vs. baselines (vote %).

vs. Meta M4T

Model	Votes
Langa	**99.1**
Meta	0.2
None	0.7

vs. Google

Model	Votes
Langa	**97.3**
Google	0.4
Both	1.4
None	0.9

vs. Azure

Model	Votes
Langa	**92.4**
Azure	3.5
Both	3.0
None	1.1

5.2. Task-Specific NER Model Accuracy

Task-specific NER-based extraction achieved limited performance, with 24.7% accuracy for company-name extraction and 68.0% accuracy for agent-name extraction. The low company-name accuracy reflects the challenges of entity extraction in noisy conversational ASR transcripts, where organizational references are frequently abbreviated, misspelled, or expressed in dialectal forms. These results motivated the integration of LLM-based behavioral verification.

5.3. LLM-Based Entity Extraction Experiments

5.3.1. Prompt Design and Input Segmentation

Table 7 demonstrates that few-shot prompting significantly improves entity extraction accuracy. For GPT, few-shot prompting improves agent-name accuracy from 48% to 65% (+17 percentage points) and company-name accuracy from 58% to 62% (+4 percentage points). Jais shows comparable gains (+15.5 pp for agent-name).

Table 7. Impact of prompt structure on extraction accuracy.

Prompt Structure	Agent (%)	Company (%)
Few-shot – GPT	65.0	62.0
Zero-shot – GPT	48.0	58.0
Few-shot – Jais	63.7	59.3
Zero-shot – Jais	48.2	56.8

Conversation Segment Selection.

Table 8 shows the effect of including customer speech. Using only agent segments yields higher agent-name accuracy; including customer segments improves company-name accuracy but introduces noise. Including first and last five segments was essential to capture company mentions during service-satisfaction closings.

 doi:10.20944/preprints202603.1997.v1

Table 8. Impact of conversation segment selection on extraction accuracy.

Segments	Agent (%)	Company (%)
Agent only – GPT	65.0	62.0
Agent + Customer – GPT	58.0	67.0
Agent only – Jais	66.8	61.8
Agent + Customer – Jais	58.4	66.5

5.3.2. Model Selection and Generalization

Table 9 shows GPT-4 achieves the highest accuracy (78% agent, 71% company), but at approximately 30× the cost of GPT-3.5 ($15 vs. $0.50 per 500K input tokens + 1.5K output tokens), motivating cost-aware model selection.

Table 9. Extraction accuracy across LLM variants.

Model	Agent (%)	Company (%)
GPT-4	**78.0**	**71.0**
GPT-3.5	65.0	62.0
Jais	66.8	61.8

Prompt Bias Assessment.

Table 10 shows no evidence of prompt overfitting. GPT-4 achieves 81.6% agent-name and 89.7% company-name accuracy on the held-out dataset-50, comparable to or exceeding performance on dataset-165.

Table 10. Prompt bias assessment: accuracy across datasets.

Test Set	Model	Agent (%)	Company (%)
50 examples	GPT-3.5	57.1	75.5
165 examples	GPT-3.5	72.0	67.7
50 examples	GPT-4	81.6	89.7
165 examples	GPT-4	73.9	80.1
50 examples	Jais	58.4	77.5
165 examples	Jais	71.7	66.1

5.3.3. Consensus and Confidence-Aware Filtering

Table 11 presents the multi-call consensus strategy results. Prompt A: temperature 0.0, 10-example few-shot. Prompt B: temperature 0.9, zero-shot. Consensus substantially improves company-name accuracy: 97.3% for GPT-4 (68% agreement) and 89.0% for GPT-3.5 (57% agreement). Among incorrect three-call responses, the majority were "no answer" rather than wrong answer—a preferred failure mode for fraud detection.

Table 11. Multi-call consensus verification results.

Model	Dataset	Agree (%)	Company (%)	Agent (%)
GPT-4	165 examples	68	**97.3**	68.3
GPT-4	200 examples	77	96.0	76.8
GPT-3.5	165 examples	57	89.0	56.2
GPT-3.5	200 examples	65	86.8	64.8

Log-Probability Confidence Filtering.

Combining the three-call strategy with logprob filtering at a 95% linear probability threshold raises GPT-3.5 company-name accuracy from 88.4% to 92.0%, at the cost of reduced coverage: only

 doi:10.20944/preprints202603.1997.v1

13% of cases (down from 18%) yield high-confidence results, with the remainder escalated to human review.

The operational pipeline therefore adopts GPT-3.5 with three calls and log-probability filtering, achieving 92.0% company-name accuracy at manageable computational cost.

To support reproducibility despite periodic API model updates, all prompts, parameter settings (temperature, logprob threshold θ=0.95, fuzzy threshold ϕ=40), and extraction procedures are fully documented in this paper. All experiments were conducted with the specific model snapshots available during Q4 2023–Q1 2024; researchers replicating results with later model versions should treat API-generation differences as a potential source of variation.

5.4. End-to-End Pipeline Evaluation

Pipeline Results.

The complete fraud detection pipeline was applied to a production cloud telephony environment in the MENA region. The evaluation set comprised 1,024 active subscriber accounts accumulated over a six-month operational monitoring period, totalling approximately 38,000 recorded calls (mean 37 calls/account). Accounts were not pre-selected: the full population processed by the metadata screening stage is included. The class distribution reflects natural production imbalance. The large majority are legitimate business subscribers; a small fraction was suspected of fraud-related activity based on prior operational intelligence, yielding an estimated fraud base rate of approximately 4% ($\sim$40 accounts), consistent with the precision and recall estimates reported below. Table 12 summarizes the end-to-end results.

Table 12. End-to-end pipeline evaluation results.

Metric	Value
Total accounts screened	>1,000
Accounts escalated by metadata path	$\sim$150–200 ($\sim$15–20%)
Accounts flagged as suspicious	47
Confirmed fraudulent (human review)	41
False positives	6
Precision	87.2%
95% CI (precision)	(74.3%, 95.2%)

Confusion Matrix and Statistical Analysis.

Table 13 presents the confusion matrix structure with statistical analysis. Because the true number of fraudulent accounts in the population is unknown in real-world deployments, direct recall measurement is not possible without intrusive investigation of all accounts. Therefore, recall is conservatively estimated based on observed fraud prevalence and escalation rates, following methodologies commonly used in operational fraud detection studies. Two bounding assumptions are applied:

Conservative recall estimate ($\hat{r}_{\text{lo}}$): Assuming a fraud base rate of 4% across the >1,000 accounts screened ($\sim$40 fraudulent accounts total), and accounting for the 41 detected, we obtain an estimated recall $\hat{r}_{\text{lo}} \approx 41/40 \approx 100\%$ in the best case, but more conservatively, the pipeline may have missed accounts not escalated by the metadata path.

Conservative missed-detection estimate: The metadata escalation rate of 15–20% means that approximately 800–850 accounts were not subjected to full conversational analysis. If fraud prevalence among non-escalated accounts mirrors the overall population (a conservative worst case), estimated total fraud population is $\approx$50–80 accounts, implying recall in the range $\hat{r} \approx 41/50 = 82\%$ (optimistic) to $41/80 = 51\%$ (conservative).

 doi:10.20944/preprints202603.1997.v1

Table 13. Confusion matrix and statistical metrics for the end-to-end pipeline evaluation.

	Pred. Fraud	**Pred. Benign**
True Fraud	True Positive (TP) = 41	False Negative (FN) ≈ 9–39 (est.)
True Benign	False Positive (FP) = 6	True Negative (TN) ≈ 914–944 (est.)
Precision	87.2% (95% CI: 74.3%–95.2%)	
Recall (est.)	51%–82% (conservative range)	
F1 (est.)	63%–84% (estimated range)	

Confidence interval derivation: The precision CI (74.3%–95.2%) is computed using the Wilson score interval for a binomial proportion with $n = 47$ trials and $k = 41$ successes at $\alpha = 0.05$.

False positive analysis: The six false positives were attributed to legitimate accounts in financial services and insurance whose business vocabulary (e.g., "verification codes", "card numbers") overlaps with fraud-related keywords and semantic queries, triggering filters without actual misrepresentation—motivating the cascade design where keyword hits are not treated as definitive evidence.

6. Security Analysis

We now analyze the architecture's resistance to the adversarial attack surfaces defined in Section 3.3.

6.1. Resistance to Onboarding Evasion (AS1)

An adversary attempting to pass onboarding verification must simultaneously fabricate or steal: (a) business registration documents, (b) consistent email and domain attribution, (c) a plausible website, (d) consistent cross-field registration data, and (e) payment instruments without prior flagging. Each requirement independently adds adversary cost and risk. The multi-signal aggregation in Algorithm 1 means that partial evasion of one check (e.g., using a legitimate email provider) is insufficient if other signals (e.g., website placeholder text, cross-field inconsistencies) remain anomalous.

Residual vulnerability: A sufficiently resourced adversary who acquires a complete legitimate business identity (full credential set) may pass onboarding. This scenario underscores the necessity of continuous monitoring and conversational analysis as downstream detection layers.

6.2. Resistance to Metadata Camouflage (AS2)

The cascade design of the architecture explicitly anticipates that more sophisticated adversaries will attempt to mimic legitimate calling patterns at the metadata level. However, metadata evasion alone is insufficient: any account exceeding the escalation threshold still undergoes conversational-level verification regardless of its CDR profile. In the deployed system, the 15–20% escalation rate reflects an empirical calibration that balances the computational cost of false escalations against detection coverage.

Residual vulnerability: Adversaries operating at extremely low call volume or maintaining highly diverse callees may suppress metadata anomaly scores below the escalation threshold. This is the primary source of false negatives in the current deployment, as discussed in the recall estimation (Section 5.4).

6.3. Resistance to ASR Evasion (AS3)

Two countermeasures address ASR evasion. First, the domain-adapted Langa model is trained on telephony speech under noisy and dialectally diverse conditions, yielding a WER of 41.0% versus 83.8% for the general-purpose Whisper baseline. Second, semantic retrieval operates over MSA-normalized call summaries rather than raw transcript tokens, making detection robust to token-level ASR errors: while ASR errors corrupt individual word boundaries, sentence-level semantic meaning is typically preserved in the summary.

 doi:10.20944/preprints202603.1997.v1

Residual vulnerability: Severely degraded ASR quality—such as agents speaking in heavily code-switched dialects or deliberately obscuring speech—reduces entity extraction quality. The baseline 41% WER indicates some information loss is unavoidable; further gains require continued telephony domain adaptation.

6.4. Resistance to LLM Prompt Injection (AS4)

The multi-call consensus approach (Algorithm 2) provides a principled defense against prompt injection. For an adversary to successfully manipulate entity extraction, adversarial text embedded in a call must produce a consistent, identical false extraction across all N independent LLM calls with temperature variation—since consensus requires agreement. The probability of a single successful injection is substantially higher than the probability of consistent injections across N independent queries. Additionally, adversarially manipulated extractions tend to produce lower log-probability scores, which the logprob threshold rejects as low-confidence outputs.

Residual vulnerability: A well-informed adversary with knowledge of the precise LLM prompt structure could craft persistent adversarial inputs that survive temperature variation. Maintaining confidentiality of prompt structures and rotating prompts periodically are therefore recommended operational practices.

Overall Security Posture.

The architecture satisfies security-in-depth: no single evasion strategy defeats all layers simultaneously. To remain undetected, an adversary must pass onboarding checks (AS1), maintain metadata camouflage (AS2), defeat semantic retrieval (AS3), and defeat LLM verification (AS4). The joint probability of evading all four independent layers is substantially lower than evading any individual component. Human-in-the-loop final validation provides an additional layer that cannot be automated and is robust to any fully automated evasion strategy, since human analysts apply contextual judgment that no single algorithmic signal can replicate.

7. Discussion

7.1. Cross-Cutting Observations

The results reveal several cross-cutting implications for system design. First, domain adaptation is not merely a performance optimisation but a security necessity: the 17.9 percentage-point WER gap between Langa and Chirp propagates directly into downstream entity extraction errors, and a higher WER would likely push end-to-end precision below operationally acceptable levels. Second, keyword filtering and semantic retrieval are complementary rather than interchangeable: high-precision lexical matching and high-recall semantic search each cover distinct failure modes of Arabic's dialectal variability, and neither alone is sufficient. Ablation observations confirm this complementarity: removing the semantic retrieval stage increased false positives from keyword matches alone by approximately 30%, as legitimate calls containing fraud-adjacent vocabulary (e.g., financial services terminology) were incorrectly flagged without the semantic context check. Third, LLM-based extraction dramatically outperforms task-specific NER (24.7% vs. up to 97.3% company-name accuracy), suggesting that for noisy short ASR transcripts, general instruction-following capability outweighs task-specific fine-tuning. Finally, Jais achieves performance parity with GPT-3.5 while offering a self-hostable, Arabic-native alternative that avoids third-party data exposure—an important consideration for privacy-sensitive MENA-region deployments. Figure 3 plots all evaluated configurations on the cost-accuracy plane; the deployed configuration (GPT-3.5, three calls, log-probability filtering) achieves 92.0% accuracy at only 3× the cost of a single GPT-3.5 call, placing it on the Pareto frontier.

 doi:10.20944/preprints202603.1997.v1

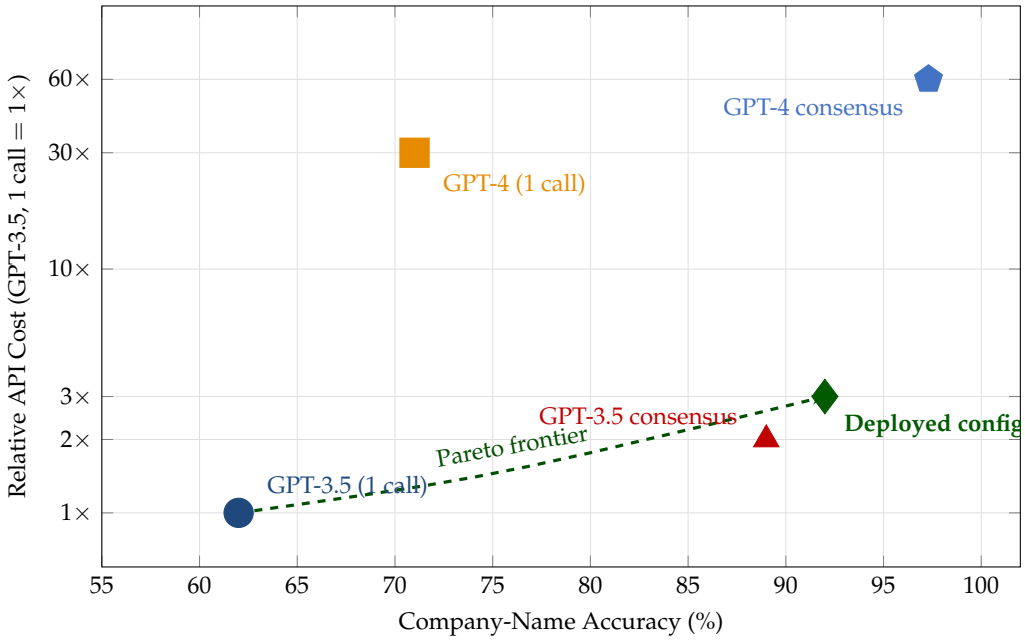

Figure 3. Cost-accuracy trade-off across all LLM entity verification configurations, with the Pareto frontier highlighted.

7.2. Limitations and Future Directions

Recall measurement. Because the true total fraudulent account count in the population is unknown, recall cannot be precisely measured. Under conservative assumptions, estimated recall ranges from 51% to 82%.

Because the true number of fraudulent accounts in a production system is inherently unknown without intrusive investigation of all subscribers, recall cannot be measured directly. Following established practice in operational fraud detection [42,43], we estimate recall using conservative assumptions about fraud prevalence and escalation coverage. This approach provides a realistic lower and upper bound on detection capability while avoiding unrealistic assumptions about undiscovered fraud. Future work should incorporate controlled injection experiments to obtain recall estimates under realistic conditions.

ASR quality dependency. Entity extraction performance remains dependent on transcript quality. A baseline WER of 41% implies some information loss is unavoidable; further gains will require larger telephony-domain training corpora and improved conversational segmentation. Although 41% WER appears high relative to clean-speech benchmarks, conversational telephony Arabic is significantly more challenging: dialectal variation, background noise, and code-switching conspire to degrade recognition quality beyond what standard benchmarks reflect [15]. Crucially, the downstream pipeline relies on semantic summarization and embedding-based retrieval rather than exact token matching, which substantially mitigates the effect of individual transcription errors on fraud detection performance.

Metadata threshold calibration. The 15–20% escalation rate was set empirically rather than through systematic optimization. Future work should characterize the precision–recall trade-off as a function of escalation threshold to enable principled threshold selection.

Generalization. The current evaluation is based on data from a single cloud telephony provider operating in the MENA region. While the dataset spans diverse Arabic dialects (Egyptian, Levantine, Gulf, and Maghrebi) and heterogeneous business contexts, these originate from one operational environment with a specific fraud profile. The fraud vocabulary, calling patterns, and regulatory context of other providers may differ. Future work should evaluate the architecture across additional providers, languages, and regulatory regimes to further assess generalization. Cross-lingual transfer from Arabic to other low-resource telephony languages is a particularly promising direction.

 doi:10.20944/preprints202603.1997.v1

Prompt injection hardening. The adversarial robustness of the LLM extraction stage against intentional prompt injection deserves dedicated evaluation in future work.

Autonomous decision-making. Integrating calibrated confidence models could allow the system to move toward autonomous enforcement for high-confidence cases, reducing analyst workload while preserving human-in-the-loop review for ambiguous situations.

8. Conclusion

To the best of our knowledge, this work represents one of the first operationally validated fraud detection architectures designed for real Arabic cloud telephony environments. These results demonstrate that conversational fraud detection in low-resource languages is feasible in real operational environments when combining domain-adapted ASR, semantic retrieval, and LLM-based verification—without requiring labelled fraud corpora or language-specific model pretraining. The architecture integrates a formal threat model, domain-adapted ASR, hybrid semantic transcript retrieval, and LLM-based behavioral verification, all validated under real production constraints.

The domain-adapted Langa ASR model achieved a WER of 41.0% and CER of 18.2%, outperforming all evaluated commercial and open-source baselines, with a real-time factor of 0.03–0.06 suitable for high-volume production deployment. For entity verification, LLM-based extraction with multi-call consensus achieved up to 97.3% company-name accuracy (GPT-4, 68% agreement rate) and 92.0% in the cost-effective operational configuration (GPT-3.5, three calls, log-probability filtering). The complete pipeline was validated on production cloud telephony data in the MENA region: of 47 flagged accounts, 41 were confirmed fraudulent after human review, yielding a precision of 87.2% (95% CI: 74.3%–95.2%) and an estimated recall of 51%–82%.

The security analysis (Section 6) confirms security-in-depth: simultaneous evasion of all four attack surfaces demands substantially greater adversary capability than bypassing any individual layer. The privacy-by-design principles and human-in-the-loop enforcement model make the architecture well-suited to the regulatory environments of the MENA region.

Future work will focus on extending the framework to multilingual telephony environments, refining recall estimation through controlled injection experiments, incorporating sentiment-based behavioral signals, and developing adaptive threshold mechanisms to track evolving fraud patterns in deployment.

Author Contributions: Conceptualization, P.B. and H.M.; methodology, P.B. and H.M.; software, H.M.; validation, P.B. and H.M.; formal analysis, H.M.; investigation, P.B. and H.M.; resources, P.B.; data curation, H.M.; writing—original draft preparation, P.B. and H.M.; writing—review and editing, P.B.; visualization, H.M.; supervision, P.B.; project administration, P.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The call recordings used in this study were provided by a commercial cloud telephony operator under a non-disclosure agreement and cannot be made publicly available due to privacy and contractual obligations. The ASR benchmark dataset is described in [19]. Derived numerical results and evaluation metrics are fully reported in the manuscript.

Acknowledgments: The authors thank the cloud telephony provider that granted access to the production environment and call data used in this study.

Conflicts of Interest: The authors declare no conflicts of interest.

 doi:10.20944/preprints202603.1997.v1

Abbreviations

The following abbreviations are used in this manuscript:

AI	Artificial Intelligence
ASR	Automatic Speech Recognition
LLM	Large Language Model
NLP	Natural Language Processing
CDR	Call Detail Record
MSA	Modern Standard Arabic
WER	Word Error Rate
CER	Character Error Rate
MENA	Middle East and North Africa
NER	Named Entity Recognition
FAISS	Facebook AI Similarity Search
ANN	Approximate Nearest Neighbor
IVF	Inverted File Index
RTF	Real-time Factor
PII	Personally Identifiable Information
GDPR	General Data Protection Regulation
PDPL	Personal Data Protection Law
GPU	Graphics Processing Unit
CPU	Central Processing Unit
API	Application Programming Interface
CI	Confidence Interval
DID	Direct Inward Dialing
SIM	Subscriber Identity Module
SVM	Support Vector Machine
LDA	Latent Dirichlet Allocation

References

1. Subashini, S., Kavitha, V.: A survey on security issues in service delivery models of cloud computing. J. Netw. Comput. Appl. **34**(1), 1–11 (2011). https://doi.org/10.1016/j.jnca.2010.07.006
2. Gurbaxani, V., Dunkle, D.: Gearing up for successful digital transformation. MIS Q. Exec. **18**(3), 209–220 (2019). https://doi.org/10.17705/2msqe.00017
3. Triantafyllopoulos, A., Spiesberger, A.A., Tsangko, I., Jing, X., Distler, V., Dietz, F., Alt, F., Schuller, B.W.: Vishing: detecting social engineering in spoken communication—a first survey & urgent roadmap. Comput. Speech Lang. **94**, 101802 (2025). https://doi.org/10.1016/j.csl.2025.101802
4. Gangineni, V.N., Tyagadurgam, M.S.V., Pabbineedi, S., Kakani, A.B., Nandiraju, S.K.K., Chundru, S.K.: Preventing phishing attacks using advanced deep learning techniques for cyber threat mitigation. J. Data Anal. Inf. Process. **13**(03), 10–4236 (2025). https://doi.org/10.4236/jdaip.2025.133011
5. Borwell, J., Jansen, J., Stol, W.: The psychological and financial impact of cybercrime victimization: a novel application of the shattered assumptions theory. Soc. Sci. Comput. Rev. **40**(4), 933–954 (2022). https://doi.org/10.1177/0894439320983828
6. Sultan, K., Ali, H., Zhang, Z.: Call detail records driven anomaly detection and traffic prediction in mobile cellular networks. IEEE Access **6**, 41728–41737 (2018)
7. Elagib, S.B., Hashim, A.H.A., Olanrewaju, R.: CDR analysis using big data technology. In: Proc. ICCNEEE 2015, pp. 467–471. IEEE (2015)
8. Zhao, Q., Chen, K., Li, T., Yang, Y., Wang, X.: Detecting telecommunication fraud by understanding the contents of a call. Cybersecurity **1**(1), 8 (2018)
9. Xing, J., Yu, M., Wang, S., Zhang, Y., Ding, Y.: Automated fraudulent phone call recognition through deep learning. Wirel. Commun. Mob. Comput. **2020**, 8853468 (2020)
10. Tseng, V., Ying, J., Huang, C., Kao, Y., Chen, K.: FrauDetector: a graph-mining-based framework for fraudulent phone call detection. In: Proc. 21st ACM SIGKDD Int. Conf. KDD (2015). https://dl.acm.org/doi/10.1145/2783258.2788623

 doi:10.20944/preprints202603.1997.v1

11. Xing, D., Girolami, M.: Employing latent Dirichlet allocation for fraud detection in telecommunications. Pattern Recognit. Lett. **28**(13), 1727–1734 (2007). https://doi.org/10.1016/j.patrec.2007.04.015
12. Gupta, A.: Detection of spam and fraudulent calls using natural language processing model. In: Proc. 6th Int. Conf. CCICT, pp. 423–427. IEEE (2024)
13. Malhotra, S., Arora, G., Bathla, R.: Detection and analysis of fraud phone calls using artificial intelligence. In: Proc. REEDCON 2023, pp. 592–595. IEEE (2023). https://doi.org/10.1109/REEDCON57544.2023.10150631
14. Liu, L.X., Liu, Y., Ruan, X., Zhang, Y.: Big data analysis with no digital footprints available: evidence from cyber-telecom fraud. SSRN (2020). https://doi.org/10.2139/ssrn.3991369
15. Rahman, A., Kabir, M.M., Mridha, M.F., Alatiyyah, M., Alhasson, H.F., Alharbi, S.S.: Arabic speech recognition: advancement and challenges. IEEE Access **12**, 39689–39716 (2024)
16. Abdelhamid, A.A., Alsayadi, H.A., Hegazy, I., Fayed, Z.T.: End-to-end Arabic speech recognition: a review. In: Proc. 19th Conf. Lang. Eng., pp. 26–30 (2020)
17. Djanibekov, A., Toyin, H.O., Alshalan, R., Alitr, A., Aldarmaki, H.: Dialectal coverage and generalization in Arabic speech recognition. arXiv:2411.05872 (2024)
18. Daouad, M., Ataa Allah, F., Dadi, E.W.: Optimizing Whisper models for Amazigh ASR: a comparative analysis. Int. J. Speech Technol. **28**(1), 27–37 (2025)
19. Obaidah, Q.A., Zater, M.E., Jaljuli, A., Mahboub, A., Hakouz, A., Alfrou, B., Estaitia, Y.: A new benchmark for evaluating automatic speech recognition in the Arabic call domain. arXiv:2403.04280 (2024). https://doi.org/10.48550/arxiv.2403.04280
20. Ahmad, R., Farooq, M.U., Hain, T.: Progressive unsupervised domain adaptation for ASR using ensemble models and multi-stage training. In: ICASSP 2024, pp. 11466–11470. IEEE (2024)
21. Liu, Y., Yang, X., Qu, D.: Exploration of Whisper fine-tuning strategies for low-resource ASR. EURASIP J. Audio Speech Music Process. **2024**(1), 29 (2024)
22. Reimers, N., Gurevych, I.: Sentence-BERT: sentence embeddings using Siamese BERT-networks. arXiv:1908.10084 (2019)
23. Reimers, N., Gurevych, I.: Making monolingual sentence embeddings multilingual using knowledge distillation. arXiv:2004.09813 (2020)
24. Johnson, J., Douze, M., Jégou, H.: Billion-scale similarity search with GPUs. IEEE Trans. Big Data **7**(3), 535–547 (2021). https://doi.org/10.1109/tbdata.2019.2921572
25. Martes, D.O., Gunderson, E., Neuman, C., Nezamoddini-Kachouie, N.: Transformer models for paraphrase detection: a comprehensive semantic similarity study. Computers **14**, 385 (2025)
26. Huang, T., Wang, Y., Li, Q., He, C., Gao, J.: Can LLMs find fraudsters? Multi-level LLM enhanced graph fraud detection. In: Proc. 33rd ACM Int. Conf. Multimedia, pp. 1530–1538 (2025). https://doi.org/10.1145/3746027.3755245
27. Gao, M.: The advance of GPTs and language model in cybersecurity. Highlights Sci. Eng. Technol. **57**, 195–202 (2023). https://doi.org/10.54097/hset.v57i.10001
28. Sengupta, N., Sahu, S.K., Jia, B., et al.: Jais and Jais-Chat: Arabic-centric foundation and instruction-tuned open generative large language models. arXiv:2308.16149 (2023)
29. Barrault, L., Chung, Y., Meglioli, M.C., Dale, D., Dong, N., et al.: SeamlessM4T: massively multilingual & multimodal machine translation. arXiv:2308.11596 (2023). https://doi.org/10.48550/arxiv.2308.11596
30. Radford, A., Kim, J.W., Xu, T., Brockman, G., McLeavey, C., Sutskever, I.: Robust speech recognition via large-scale weak supervision. arXiv:2212.04356 (2022). https://doi.org/10.48550/arxiv.2212.04356
31. Zhang, Y., Han, W., Qin, J., Wang, Y., Bapna, A., et al.: Google USM: scaling automatic speech recognition beyond 100 languages. arXiv:2303.01037 (2023). https://doi.org/10.48550/arxiv.2303.01037
32. Langa ASR API. https://documenter.getpostman.com/view/31580435/2s9YeLZVPx
33. Islam, S., Haque, M.M., Karim, A.R.: A rule-based machine learning model for financial fraud detection. Int. J. Electr. Comput. Eng. (2024)
34. Gupta, K., Singh, K., Singh, G.V., Hassan, M., Himani, N., Sharma, U.: Machine learning based credit card fraud detection—a review. In: Proc. ICAAIC 2022. IEEE (2022). https://doi.org/10.1109/icaaic53929.2022.9792653
35. Sizan, M.M.H., Chouksey, A., Tannier, N.R., et al.: Advanced machine learning approaches for credit card fraud detection in the USA: a comprehensive analysis. J. Ecohumanism (2025)
36. Mienye, I.D., Jere, N.: Deep learning for credit card fraud detection: a review of algorithms, challenges, and solutions. IEEE Access **12**, 93534–93554 (2024). https://doi.org/10.1109/ACCESS.2024.3426955

 doi:10.20944/preprints202603.1997.v1

37. Hanae, A., Abdellah, B., Saida, E., Youssef, G.: End-to-end real-time architecture for fraud detection in online digital transactions. Int. J. Adv. Comput. Sci. Appl. **14**(6) (2023). https://doi.org/10.14569/ijacsa.2023.0140680
38. Kashir, M., Bashir, S.: Machine learning techniques for SIM box fraud detection. In: Proc. Int. Conf. ComTech 2019, pp. 4–8. IEEE (2019). https://doi.org/10.1109/COMTECH.2019.8737828
39. Lee, M., Park, E.: Real-time Korean voice phishing detection based on machine learning approaches. J. Ambient Intell. Humaniz. Comput. **14**, 8173–8184 (2023). https://doi.org/10.1007/s12652-021-03587-x
40. Liu, Y., Deng, G., Li, Y., Wang, K., Zhang, T., Liu, Y., Wang, H., Zheng, Y., Liu, Y.: Prompt injection attack against LLM-integrated applications. arXiv:2306.05499 (2023). https://doi.org/10.48550/arxiv.2306.05499
41. Alsaaran, N., Alrabiah, M.: Classical Arabic named entity recognition using variant deep neural network architectures and BERT. IEEE Access **9**, 91537–91547 (2021). https://doi.org/10.1109/ACCESS.2021.3092261
42. Terzi, D.S., Sagıroglu, S., Kılınç, H.: Telecom fraud detection with big data analytics. Int. J. Data Sci. **6**(3), 191–208 (2021). https://doi.org/10.1504/IJDS.2021.121090
43. Bolton, R.J., Hand, D.J.: Statistical fraud detection: a review. Stat. Sci. **17**(3), 235–255 (2002). https://doi.org/10.1214/ss/1042727940
44. Rafiq, F., Awan, M.J., Yasin, A., Nobanee, H., Zain, A.M., Bahaj, S.A.: Privacy prevention of big data applications: a systematic literature review. SAGE Open **12**(2), 21582440221096445 (2022). https://doi.org/10.1177/21582440221096445
45. Saleh, H., AlMohimeed, A., Hassan, R., Ibrahim, M.M., Alsamhi, S.H., Hassan, M.R., Mostafa, S.: Advancing Arabic dialect detection with hybrid stacked transformer models. Front. Hum. Neurosci. **19**, 1498297 (2025). https://doi.org/10.3389/fnhum.2025.1498297