FraudTrace: Verifying Fraudulent News to Prevent Online Scam Campaigns via a Multi-Agent LLM-Based System

No Author Given

No Institute Given

Abstract. The increasing sophistication of AI-generated misinformation poses a significant challenge, particularly in Vietnamese digital environments, where scam tactics are evolving rapidly. In response, this paper presents **FraudTrace**, an automated, multi-agent verification system designed to detect and explain misinformation and online fraud. Our key contributions include (1) the construction of a diverse and realistic dataset comprising 4,221 labeled samples, (2) the introduction of a URC (Understandable Response Clarity) metric for evaluating model explainability, and (3) the deployment of a modular, role-based multi-agent architecture integrating fine-tuned Large Language Models (LLMs), Retrieval-Augmented Generation (RAG), Google Search APIs, and scam detection frameworks. Experimental results indicate that fine-tuned LLMs significantly outperform zero-shot baselines, with the best-performing model achieving over 91% in both accuracy and F1-score, while avoiding invalid responses entirely. FraudTrace not only classifies input information effectively, but also maps flagged outputs to known scam patterns using a Tactics, Techniques, and Procedures (TTP)-based reasoning approach, offering transparent and trustworthy feedback. This research establishes a strong foundation for scalable, real-time misinformation verification tailored to the Vietnamese context.

Keywords: Multi Agents · Phishing Detection · Large Language Model · Fraud Campaign · Retrieval-Augmented Generation.

1 Introduction

The widespread adoption of social media and open platforms has revolutionized how users access and share information. However, these technologies have also facilitated the rapid spread of disinformation and increasingly sophisticated online scams [1] [10] [11]. Recent fraud campaigns often involve impersonation of institutions to target vulnerable users, posing serious risks to financial security and public trust [9].

In developing countries like Vietnam, the misuse of AI to generate fake news, phishing messages, and deepfake videos is alarmingly common. These threats typically appear as short, emotionally charged texts—posts, emails, or SMS—that are difficult to manually verify. Existing tools remain fragmented, lack contextual reasoning, and struggle to provide transparent, scalable solutions [4] [6] [17].

Tackling this *information disorder* requires distinguishing between content types such as misinformation (shared without intent to deceive), disinformation (shared with malicious intent), and malinformation (genuine but harmful when misused) [15] [16] [27]. Fraud campaigns increasingly exploit these forms using emotional triggers and short-form content, demanding systems that can reason contextually across platforms [23].

Recent advances in LLMs, notably GPT-4 [20], have demonstrated strong capabilities in contextual understanding, generation, and reasoning [7]. Despite these strengths, standalone LLMs are limited in real-time verification due to static training data, input constraints, and lack of integration with external tools [21]. Effective systems must therefore combine LLMs with retrieval, search, and fact-checking mechanisms to remain adaptable and interpretable [2] [12].

To meet these demands, we propose **FraudTrace**, a modular, multi-agent verification system for real-time, explainable fraud detection. Each agent specializes in a task, such as input understanding, entity validation, or credibility assessment, using tools like RAG [18], OSINT [30], and threat intelligence APIs. This architecture enables layered, evidence-driven reasoning and transparent responses, suitable for combating dynamic and multilingual disinformation across platforms. The main contributions of this work are as follows: **(1) LLM Benchmarking and Fine-Tuning (2) Multi-Agent Architecture (3) User-Centered Interface Design**.

2 Related Work

The detection of fake news has evolved from early supervised machine learning models that relied on handcrafted features and benchmark datasets such as `FakeNewsNet` [22] and `LIAR` [26], to more explainable approaches like Case-Based Reasoning (CBR) [5] [19]. While CBR offers valuable interpretability, it often struggles with scalability and fails to adapt to novel and evolving fraud tactics in real-world environments. The rise of LLMs has enabled new capabilities in zero-shot learning, contextual reasoning, and multi-task performance. These strengths are further enhanced through techniques such as RAG [18] and the integration of OSINT [30] [31]. However, commercial models like GPT-4 [20], despite their strong performance, require significant computational resources, while open-source alternatives such as `LLaMA`, `Mistral`, `Qwen`, and `Gemma` often require fine-tuning for effective use, especially in low-resource languages like Vietnamese [14]. Moreover, many existing systems are monolithic in design, lacking modularity, cross-platform adaptability, and transparency in multilingual contexts. To address the challenge of generating robust datasets for fraud detection, Catch Me If You Can introduces a multi-agent simulation framework [25] where detectors and fraud agents co-evolve. This approach supports the generation of synthetic data across multiple fraud domains such as anti-money laundering, credit card fraud, and bot attacks. However, it primarily focuses on simulation rather than real-time detection or integration with external validation sources. In a complementary direction, DelphiAgent [29] leverages LLMs in an agentic framework inspired by the Delphi method. It coordinates multiple agents with distinct reasoning styles to extract evidence and reach a consensus judgment, improving transparency and factual reliability. While DelphiAgent excels in fact verification tasks, it remains tailored to structured datasets and lacks extensibility to broader fraud detection scenarios and external real-world tools.

In response to these limitations, we propose **FraudTrace**, a modular multi-agent framework designed to address fraud detection in a scalable, interpretable, and adaptable way. Each agent in FraudTrace is assigned a specialized role, from input analysis and phishing entity detection to source validation and final decision-making. By integrating LLMs with real-time tools such as the Google Search API, VirusTotal, and OSINT databases, FraudTrace delivers a flexible, cross-platform solution capable of handling diverse fraud scenarios. Unlike prior systems, FraudTrace emphasizes modularity, multilingual adaptability, and transparency, particularly in low-resource and evolving threat environments, bridging the gap between static reasoning models and real-world fraud detection needs.

3 Methodology

This section outlines the methods used to develop our information verification system. We first establish zero-shot baselines with general-purpose LLMs, then apply LoRA-based fine-tuning to improve task alignment. A hybrid RAG module is integrated for real-time evidence retrieval. Finally, we present the FraudTrace multi-agent framework and system architecture to support scalable, explainable verification.

3.1 Zero-Shot Prompting and Baseline LLM Benchmarking

To establish an initial benchmark and select candidate models for fine-tuning, we conducted zero-shot evaluations using a range of open-source LLMs, including **Mistral-7B**, **LLaMA3.2-3B**, **Qwen2.5-3B**, **Meta-LLaMA3.1-8B**, and **Gemma2-9B**. These models were selected for their strong performance in natural language understanding tasks and their relevance in recent literature.

Each model is prompted to classify Vietnamese news samples into *real* or *fake* categories, without any prior fine-tuning. This zero-shot evaluation aims to assess the generalization ability of the models based solely on their pretraining knowledge.

To systematically assess output quality, we introduced a metric called **Undecided Response Count (URC)**, which quantifies the number of responses that fall into the following categories:

- **Incorrect format:** Responses not conforming to the required classification format.

- **Vague or emotional responses:** Subjective or ambiguous output lacking objective judgment.
- **Off-topic responses:** Outputs that ignore or fail to address the task.

Models with valid, relevant outputs and low URC scores were shortlisted for fine-tuning, while those with formatting errors or poor task alignment were excluded. This process ensured a solid foundation for effective downstream adaptation.

3.2 Parameter-Efficient Fine-Tuning with LoRA and Unsloth

To adapt pre-trained LLMs for the Vietnamese fake news detection task under resource constraints, we adopt **Parameter-Efficient Fine-Tuning (PEFT)** using the **LoRA** method, integrated through the **Unsloth** framework [28].

LoRA (Low-Rank Adaptation) introduces trainable low-rank matrices into frozen transformer layers, enabling fine-tuning with significantly fewer parameters [13]. Specifically, given a base weight matrix $\mathbf{W} \in \mathbb{R}^{d\times k}$, LoRA learns an update $\varDelta\mathbf{W} = \mathbf{AB}$ with $\mathbf{A} \in \mathbb{R}^{d\times r}$ and $\mathbf{B} \in \mathbb{R}^{r\times k}$, where $r \ll \min(d, k)$. The updated weight is calculated as:

$$\mathbf{W}' = \mathbf{W} + \varDelta\mathbf{W} = \mathbf{W} + \mathbf{AB}$$

where:

- $\mathbf{W}$ is the original frozen weight matrix,
- $\mathbf{A}$ and $\mathbf{B}$ are the low-rank matrices learned during fine-tuning,
- r is a small rank that helps reduce training cost.

This approach enables efficient adaptation without modifying the original model weights.

Unsloth provides a lightweight fine-tuning pipeline with native support for LoRA and PEFT, including:

- Automatic freezing of original weights and injection of LoRA adapters into linear layers,
- Native support for JSONL datasets and HuggingFace `transformers` compatibility,
- Efficient GPU memory usage, allowing stable training of 3B–7B models on 12GB VRAM devices.

Each model is fine-tuned as a binary classifier with a consistent prompt template and supervision via cross-entropy loss. LoRA adapters are injected into key transformer components, including attention projections (`q_proj`, `k_proj`, `v_proj`, `o_proj`) and feedforward layers (`gate_proj`, `up_proj`, `down_proj`) to capture task-specific knowledge while minimizing parameter updates [3].

Training Configuration. Table 1 presents the training hyperparameters used across all fine-tuned models for fair comparison.

Table 1: Training hyperparameters for LoRA fine-tuning

Parameter	Value	Description
`max_seq_length`	2048	Maximum sequence length for long-text context.
`lora_r`	16	Rank of the low-rank matrices **A** and **B**.
`lora_alpha`	16	Scaling factor controlling the impact of the adapter.
`target_modules`	7 modules	LoRA-injected layers in attention and feedforward blocks.
`batch_size`	2	Small batch size due to limited VRAM, using gradient accumulation.
`grad_steps`	4	Gradient accumulation steps to simulate larger batch.
`epochs`	3	Number of training passes over the dataset.
`learning_rate`	2e-4	Learning rate for AdamW optimizer.
`weight_decay`	0.01	Regularization term to prevent overfitting.
`optimizer`	AdamW (8-bit)	Optimizer with memory-efficient computation.
`lr_scheduler`	Linear	Linear learning rate decay over training steps.

3.3 Retrieval-Augmented Generation via Hybrid Search

RAG is an advanced paradigm that combines the generative capabilities of LLMs with external information retrieval mechanisms to improve the accuracy, relevance, and transparency of generated outputs [18]. Rather than relying solely on the static knowledge embedded during pretraining, RAG systems dynamically retrieve contextual documents from external corpora, which are then used as input context for the generator. This mitigates hallucination and supports factual grounding, particularly important for high-stakes tasks like misinformation detection.

In the proposed system, RAG is central to verifying claims by enriching LLM responses with retrieved evidence. To enhance retrieval coverage and semantic matching, we adopt a **hybrid search strategy** that combines lexical and semantic search mechanisms. This hybrid retrieval framework ensures that both keyword-relevant and semantically related documents are considered, thereby reducing false negatives caused by vocabulary mismatch or paraphrasing.

- **Lexical search:** Implemented via BM25 keyword-based retrieval.
- **Semantic search:** Based on dense vector similarity in embedding space.

Lexical Search using BM25 The BM25 score of a document d with respect to a query q is computed as:

$$\text{BM25}(q,d) = \sum_{i=1}^{n} IDF(q_i) \cdot \frac{f(q_i,d)\cdot(k_1+1)}{f(q_i,d) + k_1 \cdot \left(1 - b + b \cdot \frac{|d|}{\text{avgdl}}\right)}$$

- $f(q_i, d)$: term frequency of q_i in document d

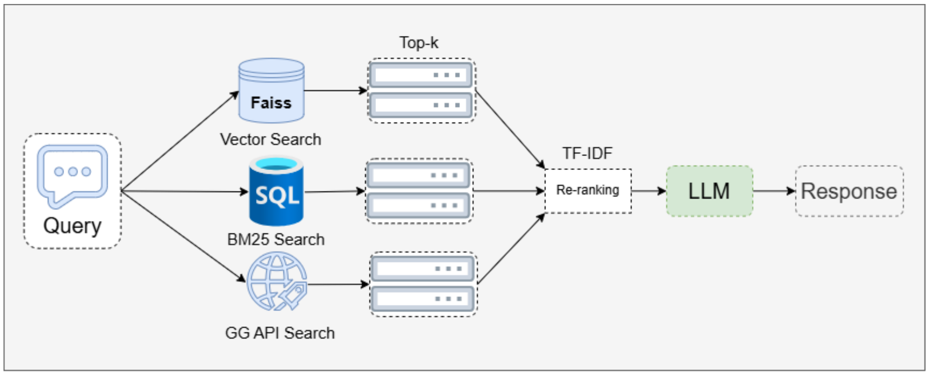


Fig. 1: Hybrid Search architecture in RAG

- $|d|$: length of document d
- avgdl: average document length in corpus
- k_1, b: hyperparameters (typically $k_1 = 1.2$, $b = 0.75$)

Semantic Search using Embedding Similarity Both query and documents are encoded as dense vectors, and semantic similarity is measured via cosine similarity:

$$\text{cosine\_similarity}(\boldsymbol{q}, \boldsymbol{d}) = \frac{\boldsymbol{q} \cdot \boldsymbol{d}}{\|\boldsymbol{q}\| \cdot \|\boldsymbol{d}\|}$$

- $\boldsymbol{q}$, $\boldsymbol{d}$: embeddings of query and document
- $\cdot$: dot product, $\|\cdot\|$: Euclidean norm

TF-IDF Re-ranking TF-IDF for term t in document d is computed as:

$$\text{TF-IDF}(t, d) = \text{TF}(t, d) \cdot \log\left(\frac{N}{DF(t)}\right)$$

- $TF(t, d)$: term frequency
- N: total number of documents
- $DF(t)$: number of documents containing t

Hybrid Scoring and Ranking The system uses weighted fusion of scores from lexical and semantic retrieval:

$$\text{Score}_{\text{hybrid}}(q, d) = \alpha \cdot \text{score}_{\text{vec}}(q, d) + (1 - \alpha) \cdot \text{score}_{\text{BM25}}(q, d)$$

- α: weighting factor ($\alpha = 0.5$ in our setup)
- $\text{score}_{\text{vec}}$: cosine similarity score
- $\text{score}_{\text{BM25}}$: normalized BM25 score

Top documents by hybrid score are further re-ranked using TF-IDF to select the most relevant ones for input to the generator.

Hybrid Search Algorithm

Input: User query q, BM25 index I_{BM25}, vector index I_{vec}, fusion weight $\alpha \in [0, 1]$
Output: Ranked document list D_{ranked}

```
1. Encode query: $\boldsymbol{q} \leftarrow \text{Embed}(q)$
2. Retrieve from BM25: $S_{BM25} \leftarrow \text{Retrieve}(q, I_{BM25})$
3. Retrieve from vector index: $S_{vec} \leftarrow \text{Retrieve}(\boldsymbol{q}, I_{vec})$
4. Initialize score dictionary $D_{\text{score}} \leftarrow \emptyset$
5. For each document $d \in S_{BM25} \cup S_{vec}$:
     – $s_{BM25} \leftarrow \text{BM25Score}(d)$
     – $s_{vec} \leftarrow \text{VectorScore}(d)$
     – $D_{\text{score}}[d] \leftarrow \alpha \cdot s_{vec} + (1 - \alpha) \cdot s_{BM25}$
6. Sort $D_{\text{score}}$ in descending order
7. Return re-ranked documents $D_{ranked}$
```

3.4 FraudTrace Agent Overview

To address the complexity and multifaceted nature of online misinformation, we propose a multi-agent architecture named **FraudTrace** built upon the CrewAI framework [8]. This architecture decomposes the end-to-end verification pipeline into specialized agents, each responsible for a clearly defined subtask. Such modularity not only facilitates parallelism and scalability but also enhances system transparency and maintainability. In addition, it overcomes the limitations of monolithic verification systems.

The overall architecture, illustrated in Figure 2, consists of six core agents, each designed to handle a distinct stage in the information verification process. By distributing responsibilities across task-specific agents, the system ensures more focused reasoning, traceable decision-making, and flexible integration with external tools and APIs.

a) Input Analyzer This agent acts as the entry point of the system, accepting user input in text or image form. For screenshots or social media posts, optical character recognition (OCR) is applied to extract textual content. The extracted text is then summarized, and key entities such as URLs, emails, phone numbers, and keywords are identified. This preprocessing standardizes noisy user input, enabling downstream agents to operate on structured and relevant information.

b) Entity Checker The Entity Checker validates the trustworthiness of extracted entities using specialized APIs. This includes:

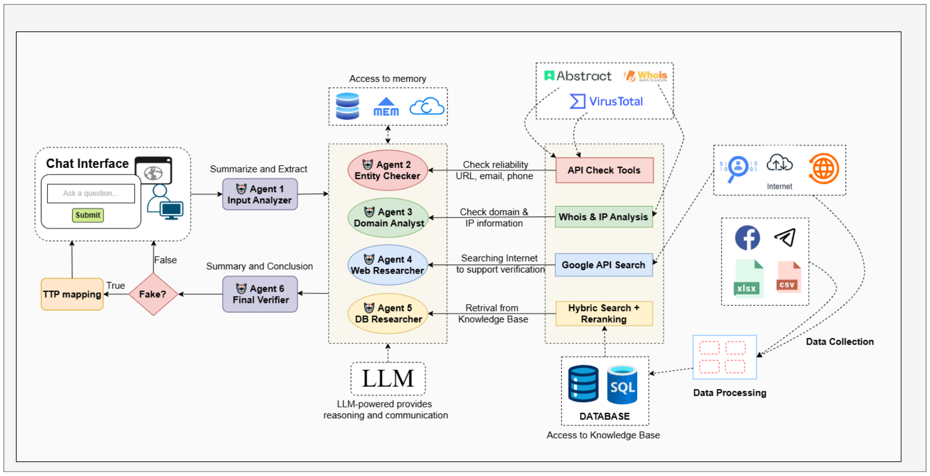


Fig. 2: The Multi-Agent Architecture of FraudTrace

- **URL analysis** via *VirusTotal API*, which aggregates threat detection results from over 70 antivirus engines to classify URLs as malicious, suspicious, or harmless.
- **Email and phone verification**: via *Abstract API*, which checks for disposable, free, or suspicious domain usage and identifies high-risk phone numbers based on format, location, and carrier metadata.

By leveraging these APIs, the system achieves greater **transparency and reliability** in entity assessment, reducing false positives and exposing early signs of phishing or fraud.

c) Domain Analyst This agent is responsible for conducting in-depth technical analysis of domain names and IP addresses. It performs **WHOIS lookups** and queries reputable IP intelligence sources to extract key metadata, including:

- Domain registration date and expiration.
- WHOIS privacy settings (anonymous vs. public ownership).
- Registrant organization and domain registrar.
- IP geolocation and hosting provider.

This information is crucial in identifying suspicious patterns, such as newly registered domains, obfuscated ownership, or hosting in high-risk regions—that are often associated with phishing or coordinated scam campaigns. By flagging such indicators, the Domain Analyst enhances the system's ability to **preemptively detect and explain technical signs of deception**, serving as an early-warning layer before content-based reasoning is applied.

d) Web Researcher Disinformation often propagates faster than static datasets can keep up. To counter this, the Web Researcher performs real-time information gathering using *Google Search API*. It retrieves up-to-date data from reputable online sources, such as official government websites, major news outlets, and filters out irrelevant or noisy content (e.g., ads, HTML tags). This agent plays a critical role in adapting the system to **rapidly evolving scams and emerging fake news trends**.

e) Database Researcher To complement live web search, the Database Researcher queries a curated internal knowledge base of cleaned and verified documents. It employs a **Hybrid Retrieval Strategy**, a combination of BM25 and semantic search (MiniLM embedding) followed by TF-IDF-based reranking to select the most contextually relevant evidence. This agent ensures high quality grounding by surfacing **trusted historical records and verified articles**.

f) Final Verifier This central reasoning agent synthesizes results from all upstream agents, structured entities, open web evidence, and internal database documents to issue a final trustworthiness verdict. Powered by GPT-4o, it performs contextual inference and delivers natural language justifications for its conclusions. This interpretability makes the system auditable, allowing end users to understand not just the verdict but the reasoning behind it.

Additionally, in cases flagged as high-risk, the Final Verifier initiates a Tactics, Techniques, and Procedures (TTP) Mapping step. This maps the observed patterns to known fraud playbooks, enabling proactive detection of campaign-level disinformation efforts.

Summary: The FraudTrace multi-agent architecture brings several critical advantages to the problem of information verification:

- **Specialization**: Each agent is fine-tuned for a specific task, improving performance and reasoning precision.
- **Transparency**: Intermediate results and modular steps enable explainable and debuggable decision flows.
- **Real-time responsiveness**: Web integration ensures adaptability to fast-changing fake news dynamics.
- **Extendability**: The architecture is modular and API-friendly, allowing easy addition of new tools or agents.

This design aligns well with the functional requirements of real-world misinformation detection systems, ensuring both robustness and scalability across diverse input scenarios.

3.5 System Software Design

To support user interaction and backend reasoning, the FraudTrace system includes both a modular API layer and a lightweight user interface (UI) designed for transparency, usability, and real-time feedback.

API Architecture for Fact Verification A set of RESTful APIs was developed to operationalize the multi-agent reasoning pipeline, enabling streamlined interactions between frontend, processing agents, and storage components. The core APIs handle input processing, content crawling, TTP database management, semantic indexing, and final verification tasks. Table 2 summarizes the main endpoints.

Table 2: Main REST API Endpoints of the System

Method	Endpoint	Description
POST	/add_news	Submit new content to the system
POST	/add_ttps	Add TTP entries to database
POST	/add_ttps_form_file	Import TTPs from CSV/Excel
POST	/ttp_embeddings	Generate and store TTP embeddings
DELETE	/delete_news	Remove article by ID
POST	/pipeline_crawl_news	Execute crawling pipeline
POST	/verify_input	Main endpoint for multi-agent verification
GET	/get_news	Retrieve stored articles
GET	/get_ttps	Retrieve TTP database entries
GET	/get_history	Retrieve user verification history
POST	/feedback	Submit user feedback for improvement

Among them, the `/verify_input` endpoint plays a central role in orchestrating the verification flow. Once invoked, the input is parsed and dispatched to corresponding AI agents for semantic analysis, knowledge retrieval, and risk assessment.

User Interface Components The system UI is designed for simplicity and functional clarity. It offers key components to guide users through input submission, verification, and feedback:

- **Input Box:** Accepts user provided text or image-based content.
- **Verification Trigger:** A single action button that invokes the backend API to initiate multi-agent processing.
- **Feedback Module:** Allows users to report errors, suggest corrections, or provide comments to improve system accuracy.
- **TTP Mapping View:** When suspicious content is detected, this module highlights related attack patterns (TTPs), offering descriptions and reference links to official sources. It helps clarify the attacker's main objective (Tactics), the methods used to carry it out (Techniques), and the specific procedure by which the campaign is executed (Procedures).
- **Admin Dashboard:** Grants authorized users access to monitoring tools, manual data edits, and analytics on user feedback and verification trends.

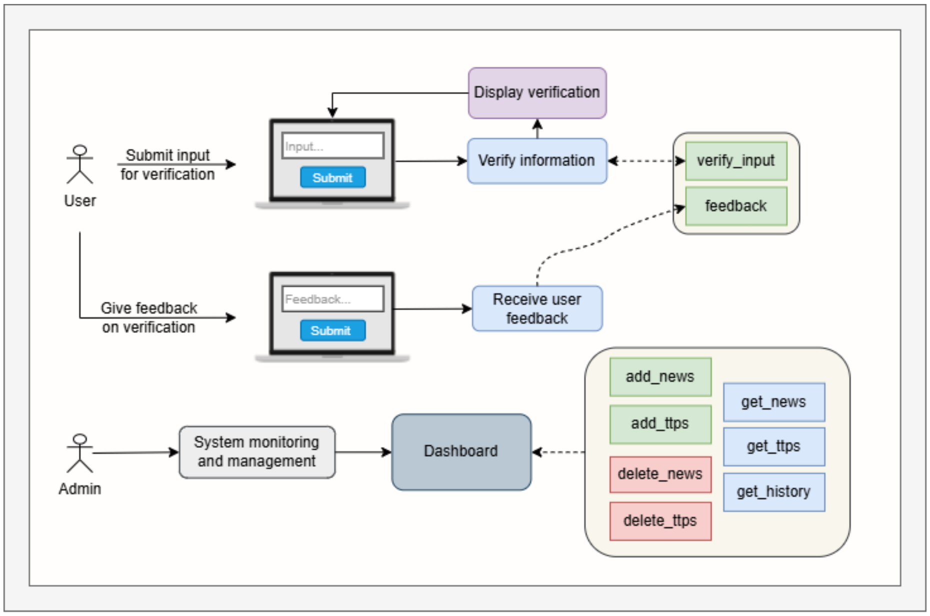


Fig. 3: System software interface and component workflow

Figure 3 shows the complete software interface and its functional components. This UI structure supports efficient user interaction while enabling transparent AI-driven decisions.

4 Experiments and Evaluation

4.1 Dataset Construction and Annotation

To support robust model training in a real-world Vietnamese context, we curated a diverse and representative dataset by combining publicly available resources with newly collected samples. Specifically, this study integrates two labeled datasets from prior research: *"Utilizing Transformer Models to Detect Vietnamese Fake News on Social Media Platforms"*[14] and *"Detecting Vietnamese Fake News"*[24]. Both datasets contain Vietnamese-language news articles categorized as real or fake, covering a broad range of topics such as politics, society, and prevalent online disinformation. These resources serve as the foundational corpus for our training process.

To enhance topic diversity and recency, we further collected original samples from various digital environments. The additional data includes disinformation content sourced from social media platforms, messaging services, and official scam alerts issued by Vietnamese authorities. These entries reflect common fraud tactics such as impersonation of government entities, phishing links, fake financial investment schemes, and fraudulent job recruitment campaigns.

All data entries were cleaned through a standard process. Labels were assigned based on trusted sources and manual review to ensure high accuracy. Fake news was identified by manipulative tone, unverifiable claims, and scam-like patterns, while real news came from reputable Vietnamese media. To improve linguistic diversity and model robustness, light data augmentation was applied—such as synonym replacement and sentence reordering—while keeping original labels intact.

The final dataset consists of **4,221 labeled instances** (2,299 fake, 1,922 real), which were split into training, validation, and test subsets. The training set supports model learning, the validation set aids hyperparameter tuning, and the test set provides objective evaluation metrics.

Table 3: Distribution of Fake vs Real Samples across Dataset Splits

Label	Total Numbers	Train	Val	Test
Fake	2,299	1,391	454	454
Real	1,922	1,143	389	390
Total	**4,221**	**2,534**	**843**	**844**

4.2 Experiment Settings and Evaluation Metrics

To evaluate model performance in fake news detection, we adopt four widely used classification metrics: **Accuracy**, **Precision**, **Recall**, and **F1-score**, which together assess both overall correctness and class-wise balance.

In addition, we introduce the URC metric to capture cases where the model fails to produce a valid classification. This metric reflects the number of test instances for which the model does not generate a valid binary label (either `real` or `fake`). URC serves as a practical indicator of output reliability, especially in real-world deployments where strict adherence to expected response formats is essential.

Table 4: Evaluation metrics.

Metrics	**Description**
True Positive (TP)	Fake news correctly predicted as fake.
True Negative (TN)	Real news correctly predicted as real.
False Positive (FP)	Real news incorrectly predicted as fake.
False Negative (FN)	Fake news incorrectly predicted as real.
Accuracy	$(TP + TN)/(TP + TN + FP + FN)$
Precision	$TP/(TP + FP)$
Recall	$TP/(TP + FN)$
F1-score (F1)	2*((Pre*Rec)/(Pre + Rec))
URC	Count of invalid model responses.

4.3 Experimental Results

This section presents our experimental findings to address the following core research questions:

- **RQ1**: How effectively can LLMs verify Vietnamese news across various disinformation types?
- **RQ2**: How does fine-tuning impact the stability and accuracy of LLMs in realistic classification tasks?
- **RQ3**: How can a multi-agent framework like FraudTrace ensure real-time, reliable verification with specialized tools?
- **RQ4**: How can the system enhance user transparency by associating detected disinformation with known scam campaigns?

To enable robust evaluation of fake news detection in Vietnamese, we first curated a diverse dataset comprising both public and newly collected sources, covering multiple disinformation types and linguistic variations. This dataset supports comprehensive benchmarking and serves as the foundation for fine-tuning, directly addressing **RQ1** regarding the model's ability to handle varied misinformation contexts in Vietnamese.

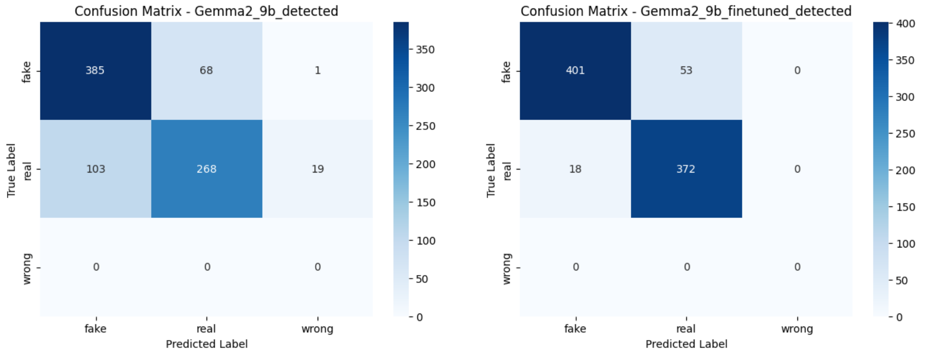
Table 5: Performance comparison before and after fine-tuning

Model	Accuracy	F1-Score	Precision	Recall	URC
Zero-shot					
Mistral-7B	0.455	0.4684	0.7081	0.455	194
Llama3.2-3B	0.6315	0.6718	0.7275	0.6315	114
Qwen2.5-3B	0.6647	0.6371	0.7623	0.6647	33
Meta-Llama3.1-8B	0.609	0.5678	0.7099	0.609	44
Gemma2-9B	0.7737	0.7809	0.7929	0.7737	20
Fine-Tuned					
Llama3.2-3B	0.8021	0.8155	0.8335	0.8021	26
Qwen2.5-3B	0.8199	0.8229	0.8448	0.8199	14
Meta-Llama3.1-8B	0.7654	0.7733	0.8394	0.7654	30
Gemma2-9B	**0.9159**	**0.9160**	**0.9193**	**0.9159**	**0**

Model Performance Before and After Fine-tuning We then evaluated a suite of LLMs under both zero-shot and fine-tuned configurations. Table 5 presents the comparative results across Accuracy, Precision, Recall, F1-score, and URC — a custom metric indicating the number of vague, incomplete, or off-topic model outputs. A lower URC score reflects better alignment with the task format and improved real-world applicability.

- **Zero-shot Evaluation**: Among the zero-shot models, **Gemma2-9B** consistently outperformed other baselines, achieving 77.37% Accuracy and 78.09% F1-score with a low URC of 20. This suggests a strong generalization ability and alignment with the Vietnamese verification task, even without task-specific tuning. In contrast, **Mistral-7B** performed the worst with only 45.5% Accuracy and a high URC of 194, indicating a substantial proportion of vague or off-topic responses. As a result, Mistral-7B was excluded from the fine-tuning phase.
- **Impact of Fine-tuning**: Fine-tuning with LoRA adapters led to substantial gains across all metrics. For instance, **Gemma2-9B** improved to 91.59% Accuracy and F1-score while reducing URC to zero, indicating not only high classification performance but also consistent output formatting. These results affirm the effectiveness of task-specific tuning in enhancing both accuracy and reliability, providing a concrete answer to **RQ2**.
- **URC as a Stability Indicator:** While conventional metrics assess classification quality, URC captures model compliance with output constraints. A high-accuracy model that frequently produces invalid or ambiguous responses may fail in real-world applications. By achieving URC = 0, the fine-tuned models demonstrate task fluency and readiness for deployment.

Figure 4 visualizes this improvement. Compared to the zero-shot confusion matrix, which shows frequent misclassification between fake and real, the fine-tuned variant achieves high precision and recall with no undecided predictions.

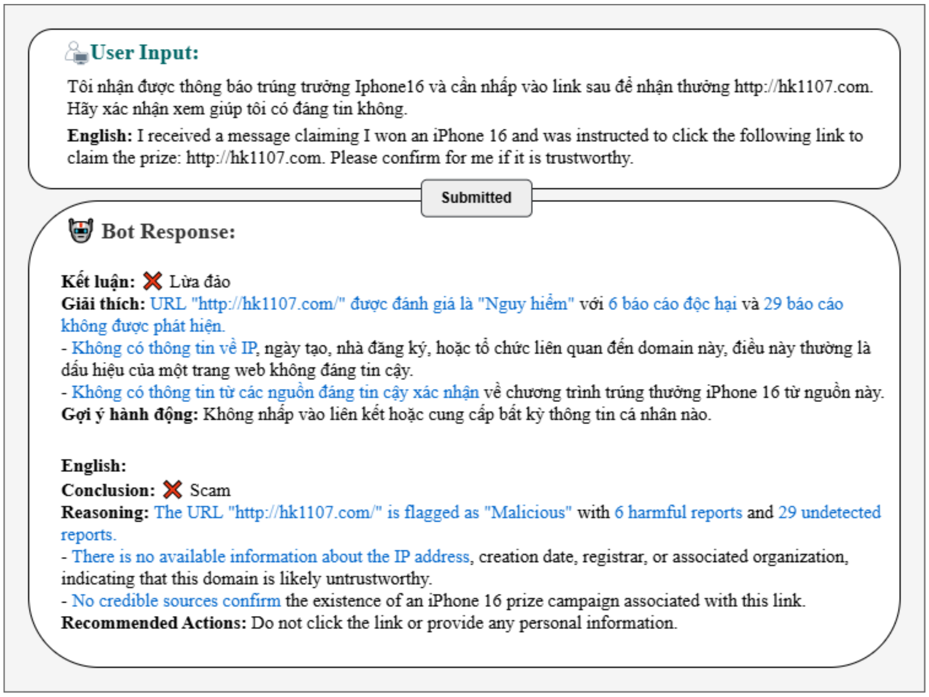


(a) Gemma2-9B (Zero-shot)

(b) Gemma2-9B (Fine-tuned)

Fig. 4: Comparison of confusion matrices before and after fine-tuning for Gemma2-9B.

Multi-agent Verification System and Transparent Reasoning To address the evolving nature of online disinformation and increasingly sophisticated scam tactics, we propose a modular verification framework named **FraudTrace**. Unlike monolithic systems, FraudTrace adopts a multiagent architecture that decomposes the verification process into specialized subtasks, such as input analysis, entity checking, evidence retrieval, and scam classification each handled by a dedicated agent. These agents operate collaboratively within a shared memory context and are dynamically orchestrated depending on input complexity.



User Input:

Tôi nhận được thông báo trúng trưởng Iphone16 và cần nhấp vào link sau để nhận thưởng http://hk1107.com. Hãy xác nhận xem giúp tôi có đáng tin không.

English: I received a message claiming I won an iPhone 16 and was instructed to click the following link to claim the prize: http://hk1107.com. Please confirm for me if it is trustworthy.

Bot Response:

Kết luận: ❌ Lừa đảo
Giải thích: URL "http://hk1107.com/" được đánh giá là "Nguy hiểm" với 6 báo cáo độc hại và 29 báo cáo không được phát hiện.
- Không có thông tin về IP, ngày tạo, nhà đăng ký, hoặc tổ chức liên quan đến domain này, điều này thường là dấu hiệu của một trang web không đáng tin cậy.
- Không có thông tin từ các nguồn đáng tin cậy xác nhận về chương trình trúng thưởng iPhone 16 từ nguồn này.
Gợi ý hành động: Không nhấp vào liên kết hoặc cung cấp bất kỳ thông tin cá nhân nào.

English:
Conclusion: ❌ Scam
Reasoning: The URL "http://hk1107.com/" is flagged as "Malicious" with 6 harmful reports and 29 undetected reports.
- There is no available information about the IP address, creation date, registrar, or associated organization, indicating that this domain is likely untrustworthy.
- No credible sources confirm the existence of an iPhone 16 prize campaign associated with this link.
Recommended Actions: Do not click the link or provide any personal information.

Fig. 5: Illustrative interface of the FraudTrace system

This decentralized agent-based design directly addresses **RQ3**, enabling the system to scale efficiently, adapt in real time, and maintain reliability across varied information verification tasks. By assigning focused responsibilities to each agent, FraudTrace enhances interpretability, allows concurrent processing, and simplifies debugging and maintenance in practical deployments.

Figure 5 illustrates the main user interface of the FraudTrace system. This interface allows users to input either text or screenshots for verification. Once submitted, the system returns a structured response including the final verdict (e.g., "Scam"), reasoning based on retrieved evidence and entity checks, and recommended user actions. This interface is designed to be user friendly and accessible to the general public.

Beyond this main interface, the system also includes:

- A **TTP Mapping View** which visually associates flagged content with known scam techniques, improving awareness and fraud literacy.
- A **Feedback Box** where users can contribute comments, corrections, or suggestions about the system's output enhancing the refinement loop.
- An internal **Admin Dashboard** used by researchers to monitor verification history, analyze user behavior, and evaluate the impact and reliability of each component for ongoing system improvement.

In response to **RQ4**, the integration of post-verification explanations and scam taxonomy mapping ensures transparency and enhances user trust. When a suspicious message is flagged, the system not only provides a clear label but also contextualizes it within broader threat patterns, such as impersonation or phishing. This approach transforms the system from a binary classifier into a comprehensible and educational tool for navigating digital misinformation.

5 Conclusion

In this paper, we introduced **FraudTrace**, an automated information verification system designed to combat the growing threat of misinformation in Vietnamese digital environments. Key contributions include the construction of a diverse training dataset, the introduction of the URC metric to assess model clarity, and the deployment of a modular multi-agent architecture for scalable and transparent verification. Experimental results indicate that fine-tuned LLMs significantly outperform zero-shot baselines, with the best-performing model achieving over 91% in both accuracy and F1-score while producing no invalid responses. FraudTrace classifies content effectively and provides explainable feedback by mapping flagged instances to known scam tactics.

While some limitations persist, particularly in distinguishing official alerts from scam messages due to data ambiguity, FraudTrace demonstrates practical value and readiness for deployment. The study paves the way for future development of scalable, explainable verification frameworks tailored to Vietnamese misinformation and fraud landscapes.

References

1. Abdillah, R., Shukur, Z., Mohd, M., Murah, T.M.Z.: Phishing classification techniques: A systematic literature review. IEEE Access **10**, 41574–41591 (2022)
2. Afane, K., Wei, W., Mao, Y., Farooq, J., Chen, J.: Next-generation phishing: How llm agents empower cyber attackers. In: 2024 IEEE International Conference on Big Data (BigData). pp. 2558–2567. IEEE (2024)
3. Ahmad, S., Khan, M., Kumari, S.: Fake news detection and classification: A comparative study of convolutional neural networks, large language models, and natural language processing models. Procedia Computer Science **218**, 2761–2770 (2023)
4. Alam, F., Cresci, S., Chakraborty, T., Silvestri, F., Dimitrov, D., Martino, G.D.S., Shaar, S., Firooz, H., Nakov, P.: A survey on multimodal disinformation detection. arXiv preprint arXiv:2103.12541 (2021)
5. Almandouh, M.E., Alrahmawy, M.F., Eisa, M., Elhoseny, M., Tolba, A.S.: Ensemble based high performance deep learning models for fake news detection. Scientific Reports (2024)
6. Azri, A., Favre, C., Harbi, N., Darmont, J., Noûs, C.: Monitor: a multimodal fusion framework to assess message veracity in social networks. In: Advances in Databases and Information Systems: 25th European Conference, ADBIS 2021, Tartu, Estonia, August 24–26, 2021, Proceedings 25. pp. 73–87. Springer (2021)
7. Bethany, M., Galiopoulos, A., Bethany, E., Karkevandi, M.B., Vishwamitra, N., Najafirad, P.: Large language model lateral spear phishing: A comparative study in large-scale organizational settings. arXiv preprint arXiv:2401.09727 (2024)
8. CrewAI: Crewai: Build agentic workflows with multi-agent collaboration (2024), crewAI Documentation
9. Desolda, G., Ferro, L.S., Marrella, A., Catarci, T., Costabile, M.F.: Human factors in phishing attacks: a systematic literature review. ACM Computing Surveys (CSUR) **54**(8), 1–35 (2021)
10. Group, A.P.W.: Phishing attack trends report – 4q 2024 (2024), `https://apwg.org/trendsreports/`, [accessed 07-June-2025]
11. Guo, Z., Cho, J.H., Chen, R., Sengupta, S., Hong, M., Mitra, T.: Safer: Social capital-based friend recommendation to defend against phishing attacks. In: Proceedings of the International AAAI Conference on Web and Social Media. vol. 16, pp. 241–252 (2022)
12. Hazell, J.: Large language models can be used to effectively scale spear phishing campaigns. arXiv preprint arXiv:2305.06972 (2023)
13. Hu, E., Shen, Y., Wallis, P., et al.: Lora: Low-rank adaptation of large language models. arXiv preprint arXiv:2106.09685 (2021)
14. Huynh, A.T., Tran, P.: Utilizing transformer models to detect vietnamese fake news on social media platforms. KSII Transactions on Internet and Information Systems **19**(2), 1234–1249 (2025), `https://itiis.org/digital-library/102085`
15. Ireton, C., Posetti, J.: Journalism, 'Fake News' and Disinformation: A Handbook for Journalism Education and Training. UNESCO Publishing (2018)
16. Jr, E.C.T., Lim, Z.W., Ling, R.: Defining "fake news": A typology of scholarly definitions. Digital Journalism (2018)
17. LekshmiAmmal, H.R., Madasamy, A.K.: A reasoning based explainable multimodal fake news detection for low resource language using large language models and transformers. Journal of Big Data **12**(1), 46 (2025)
18. Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., et al.: Retrieval-augmented generation for knowledge-intensive nlp tasks. In: Advances

in Neural Information Processing Systems. vol. 33, pp. 9459–9474 (2020), https://arxiv.org/abs/2005.11401
19. Monterrubio, S.M.M., Noain-Sánchez, A., Pérez, E.V., Crespo, R.G.: Coronavirus fake news detection via medosint check in health care official bulletins with cbr explanation (2023)
20. OpenAI: Gpt-4 technical report (2023), https://openai.com/research/gpt-4
21. Papageorgiou, E., Chronis, C., Varlamis, I., Himeur, Y.: A survey on the use of large language models (llms) in fake news. Future Internet **16**(8), 298 (2024)
22. Shu, K.: Fakenewsnet (2021), https://github.com/KaiDMML/FakeNewsNet
23. Tian, K., Jan, S.T., Hu, H., Yao, D., Wang, G.: Needle in a haystack: Tracking down elite phishing domains in the wild. In: Proceedings of the Internet Measurement Conference 2018. pp. 429–442 (2018)
24. Vinh, V.D., Do, P.: Detecting vietnamese fake news. Can Tho University Journal of Science **58**(1), 55–64 (2022), https://ctujs.ctu.edu.vn/index.php/ctujs/article/view/680/650
25. Wang, Q., Tsai, W.T., Shi, T., Liu, Z., Du, B.: Catch me if you can: A multi-agent synthetic fraud detection framework for complex networks. In: 2025 IEEE 41st International Conference on Data Engineering (ICDE). pp. 3629–3641. IEEE Computer Society (2025)
26. Wang, Y.W.: "liar, liar pants on fire": A new benchmark dataset for fake news detection. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics. pp. 422–426 (2017)
27. Wardle, C., Derakhshan, H.: Information disorder: Toward an interdisciplinary framework for research and policy making (2017)
28. Welbl, J., Stenning, S.: Fine-tuned 'small' llms (still) significantly outperform zero-shot generative ai models in text classification. arXiv preprint arXiv:2402.01325 (2024)
29. Xiong, C., Zheng, G., Ma, X., Li, C., Zeng, J.: Delphiagent: A trustworthy multi-agent verification framework for automated fact verification. Information Processing & Management **62**(6), 104241 (2025)
30. Yadav, A., Kumar, A., Singh, V.: Open-source intelligence: A comprehensive review of the current state, applications and future perspectives in cybersecurity. Cybersecurity Journal (2023)
31. Zhang, M., Lee, D., Zhao, Y.: Analysis of disinformation and fake news detection using fine-tuned large language model. arXiv preprint arXiv:2309.04704 (2023), https://ar5iv.labs.arxiv.org/html/2309.04704