



Abhimanyu Chakravyuha: An Evidence-Aware AI Framework for Fraud Intelligence Through Semantic Analysis, Mutation Detection, and Graph-Based Reasoning

MD Auranzeb Khan¹, Kunal², Srijan Mani Tripathi³, Vyom Pandey⁴, Dr.G Paavai Anand⁵

Student, Computer Science and Engineering, SRM Institute of Science and Technology, Chennai, India ¹

Student, Computer Science and Engineering, SRM Institute of Science and Technology, Chennai, India ²

Student, Computer Science and Engineering, SRM Institute of Science and Technology, Chennai, India ³

Student, Computer Science and Engineering, SRM Institute of Science and Technology, Chennai, India ⁴

Professor of the Department, Department of Computer Science and Engineering,

SRM Institute of Science and Technology, Chennai, India⁵

Abstract: Fraudulent activities increasingly involve evolving narratives, altered textual claims, interconnected entities, and coordinated patterns that cannot always be identified by examining individual records in isolation. This paper presents Abhimanyu Chakravyuha, a proposed AI-powered fraud intelligence framework integrating natural language processing, semantic similarity analysis, mutation detection, graph-based relationship analysis, and evidence-aware verification.

The framework processes textual claims and related records, retrieves semantically relevant evidence, identifies potential modifications to previously observed patterns, and represents relationships using a graph structure. An evidence-verification component evaluates whether available information supports, contradicts, or fails to establish a claim. An abstention mechanism allows the system to return an inconclusive result when evidence is insufficient or conflicting.

The design emphasizes evidence traceability, uncertainty management, and human oversight. It separates semantic similarity from factual support, suspicious patterns from confirmed fraud, and automated analysis from human-reviewed decisions. This paper specifies the architecture, methodology, and experimental design; quantitative performance is not claimed without experimental validation.

Keywords: Fraud Intelligence, Natural Language Processing, Semantic Similarity, Claim Verification, Mutation Detection, Knowledge Graphs, Explainable AI, Evidence Retrieval, Abstention, Uncertainty Calibration

I. INTRODUCTION

1. Background

The digitization of financial transactions, online communication, and commercial interactions has created opportunities for increasingly sophisticated fraud. Fraudulent behavior may involve misleading descriptions, modified statements, repeated narratives, suspicious entity relationships, or coordinated activity distributed across multiple records.

Traditional fraud detection methods commonly use predefined rules, statistical techniques, supervised classification, and anomaly detection. Although useful, these methods can struggle when known patterns change or when suspicious activity is distributed across interconnected records. Natural Language Processing (NLP) enables systems to interpret text beyond exact keyword matching: semantic representations can identify passages with similar meanings despite differences in wording, information retrieval can locate potentially relevant evidence, and graph-based representations can model relationships among entities, claims, and records.

However, similarity does not establish truth, graph connectivity does not prove wrongdoing, and a model prediction is not proof of fraud. A practical system must distinguish suspicious signals from verified evidence. Abhimanyu Chakravyuha is proposed as a modular framework combining these capabilities to transform fragmented information into evidence-linked assessments that support human investigation.



2. Key Definitions

- **Fraud Intelligence:** The structured analysis of claims, records, entities, and relationships to surface evidence-linked signals that support, but do not replace, human fraud investigation.
- **Semantic Similarity:** A measure of closeness in meaning between two texts, computed from vector embeddings (for example, cosine similarity), independent of exact wording.
- **Evidence Retrieval:** The process of ranking an evidence collection by similarity to a claim and returning the top-k candidate passages for subsequent verification.
- **Claim Verification:** Evaluating a claim against retrieved evidence and assigning one of three labels: Supported, Refuted, or Insufficient Evidence.
- **Mutation Detection:** Comparing related claims to find changes that may affect meaning, such as substituted entities, altered quantities or dates, inserted or removed negation, and changed causal statements.
- **Fraud Intelligence Graph:** A graph $G = (V, R)$ whose nodes are claims, evidence documents, people, organizations, accounts, transactions, and events, and whose edges encode typed, provenance-carrying relationships.
- **Abstention:** The system behavior of returning Insufficient Evidence, together with a recorded reason, when calibrated confidence falls below a threshold or evidence is missing, conflicting, or out of domain.
- **Evidence Traceability:** The property that every output links to identifiable evidence, graph context, mutation indicators, model version, and timestamp so that reviewers can reconstruct the analytical path.
- **Provenance:** Metadata describing the source, identifier, timestamp, and processing status of a record or relationship.

3. Problem Statement and Research Gap

Existing analytical systems may face the following limitations:

- **Lexical variation:** similar claims may use different wording, synonyms, or paraphrases.
- **Pattern mutation:** previously observed narratives may be modified to evade exact-match rules.
- **Fragmented evidence:** relevant information may be distributed across multiple documents and entities.
- **Relationship complexity:** suspicious patterns may emerge only when related records are examined together.
- **Limited explainability:** classification scores may not reveal the evidence behind a decision.
- **Uncertainty:** available information may be incomplete, outdated, contradictory, or unreliable.

The research problem is to design a framework that integrates semantic analysis, mutation detection, graph-based reasoning, and evidence-aware verification while explicitly representing uncertainty. The literature establishes the importance of each component individually; the practical value of integrating them depends on how they exchange information and how their outputs are evaluated. It is not assumed that no previous system has combined these techniques. The research question is whether this integration produces measurable improvements over simpler baselines, particularly in evidence traceability and in uncertain cases.

4. Objective

Develop Abhimanyu Chakravyuha, a unified and evidence-aware fraud intelligence framework that:

- Provides a modular NLP pipeline for processing claims and related records.
- Identifies semantically related records using embeddings and similarity-based retrieval.
- Investigates potentially meaningful textual and factual mutations.
- Represents relationships among claims, entities, records, and evidence through graph structures.
- Distinguishes supported, refuted, and inconclusive claims.
- Incorporates abstention when evidence is insufficient or conflicting.
- Is evaluated using suitable datasets, baseline systems, and ablation studies.

5. Research Questions

- **RQ1:** How effectively can semantic retrieval identify evidence when claim wording differs from the evidence?
- **RQ2:** Can mutation analysis identify meaningful changes without classifying every textual difference as suspicious?
- **RQ3:** Does graph context improve analysis beyond isolated text classification?
- **RQ4:** Can evidence-aware verification and abstention reduce unsupported conclusions?
- **RQ5:** What trade-offs arise between performance, interpretability, and computational cost?

6. Research Contributions

- An integrated architecture combining semantic retrieval, mutation analysis, graph construction, and claim verification.
- Evidence-linked assessments that connect outputs to identifiable sources.



- Mutation-aware comparison of entities, quantities, dates, and relationships.
- Uncertainty-sensitive decision-making through abstention.
- A reproducible evaluation framework for retrieval, verification, calibration, and graph analysis.

These contributions describe the proposed design. Their novelty and effectiveness must be established through experiments and comparison with existing approaches.

7. Scope (Limitations)

- Presented as a design and methodology; no experimental results are reported in this paper.
- Designed primarily for textual claims and related records, with structured metadata as supporting input.
- Verification labels describe evidence support, not fraud; fraud determination requires human-reviewed investigation.
- Generic fact-verification datasets do not represent financial fraud; fraud-specific data is required for direct evaluation.
- Coefficients, thresholds, and calibration values are design parameters that must be set using validation data.
- No integration with institutional case-management or transaction-monitoring systems is assumed.

II. LITERATURE REVIEW

1. Conventional Fraud Detection

Rule-based fraud detection systems use predefined conditions and thresholds. They are interpretable but depend on maintaining relevant rules. Statistical and machine-learning approaches learn patterns from historical data, but their performance may decline when fraud strategies change or data distributions shift. These limitations motivate the investigation of systems combining learned representations, evidence retrieval, relationship analysis, and uncertainty handling.

2. Semantic Search and NLP

Traditional information retrieval methods often rely on lexical overlap and term-weighting techniques. Dense retrieval systems represent text as numerical vectors, allowing semantically related passages to be identified using measures such as cosine similarity. Semantic retrieval is useful when a claim and an evidence passage express similar meanings with different words. However, similar passages may differ in dates, quantities, entities, or negation. Semantic similarity should therefore be used to discover evidence rather than to determine truth.

3. Automated Fact Verification

Automated fact verification evaluates a claim against an evidence collection and assigns a verification label. The FEVER dataset introduced by Thorne et al. contains 185,445 claims labelled Supported, Refuted, or NotEnoughInfo, with evidence annotations for relevant claims [1]. FEVEROUS extends this direction by incorporating structured evidence, including tables and lists, alongside unstructured text [2]. These datasets provide useful benchmarks for evidence-based claim verification. However, their labels do not directly represent financial fraud, so fraud-specific datasets are required to evaluate actual fraud detection.

4. Graph-Based Analysis

Graphs represent information as nodes connected by edges. Fraud intelligence graphs may represent accounts, organizations, transactions, documents, claims, and relationships. Graph analysis can reveal shared attributes and connected patterns that are difficult to identify through isolated classification, and research surveys have investigated graph neural networks and graph-based methods for misinformation detection [3], [4]. Nevertheless, shared attributes or dense connections do not independently establish fraud. Graph information should support investigation rather than act as proof of wrongdoing.

5. Knowledge Graphs for Fact Checking

Knowledge graphs represent entities and relationships in structured form and can support automated fact checking by connecting claims with relevant entities and evidence. Qudus et al. published a survey of fact checking over knowledge graphs in 2025 [5]. This direction motivates integrating structured relationships with textual evidence and uncertainty-aware verification.

6. Research Gap

Abhimanyu Chakravyuha investigates an architecture in which semantic retrieval discovers evidence, mutation analysis compares related claims, graph analysis organizes relationships, and evidence-aware verification assesses claims.



Whether this integration yields measurable improvements over simpler baselines, particularly in evidence traceability and uncertain cases, is the open question addressed by the proposed evaluation.

III. MATERIALS AND METHODS

The design of Abhimanyu Chakravyuha follows a modular principle: each component accepts structured inputs, produces structured outputs, and can be evaluated independently. The framework processes textual claims and related records, including document excerpts, historical records, transaction descriptions, and structured metadata. The proposed implementation is Python-based and uses sentence embeddings for semantic representation, a vector index such as FAISS for retrieval, named-entity recognition with deterministic extractors, NetworkX or a graph database for relationship analysis, and a natural language inference classifier or evidence-grounded language model for verification.

The technologies listed in this paper are suggested implementation options. The final implementation must document the embedding model, index configuration, similarity metric, retrieval parameters, and the exact technologies, versions, models, and configurations used.

1. System Overview

The framework contains six logical components, each producing outputs that can be evaluated independently:

- **Component 1:** Input processing and normalization.
- **Component 2:** Semantic representation and evidence retrieval.
- **Component 3:** Mutation and inconsistency analysis.
- **Component 4:** Entity extraction and graph construction.
- **Component 5:** Evidence-aware claim verification.
- **Component 6:** Confidence assessment, abstention, and human review.

2. Input Processing and Normalization

The system accepts textual claims and related records. Preprocessing prepares each record while preserving the distinctions on which later analysis depends.

- Text normalization and sentence segmentation.
- Entity extraction (people, organizations, accounts, amounts, dates).
- Language detection where supported.
- Assignment of record identifiers and provenance metadata.
- Preservation of negation, numerical amounts, dates, and named entities; these must not be erased during normalization.

Each record contains its identifier, text, extracted attributes, source metadata, and processing status.

Example: Record R-0412 → text: "Payment of 50,000 was approved on 12 March" → attributes: {amount: 50,000, date: 12 March, action: approved} → source: internal ledger excerpt → status: normalized. (Illustrative record structure.)

3. Processing Pipeline

Step-by-step processing flow:

- Input Ingestion (claims and related records with provenance).
- Preprocessing (normalization, sentence segmentation, entity extraction).
- Semantic Embedding (claim and evidence vectors).
- Evidence Retrieval (top-k candidates ranked by cosine similarity).
- Mutation Analysis (semantic and attribute comparison of related claims).
- Graph Construction (entity resolution, nodes, typed relationships, provenance).
- Evidence-Aware Verification (Supported, Refuted, or Insufficient Evidence).
- Confidence Assessment and Abstention (calibrated threshold with recorded reason).
- Traceable Output and Human Review (evidence-linked record for investigators).

4. Module Design

A. Semantic Representation and Retrieval

Let a claim be represented by c and the evidence collection by $E = \{e_1, e_2, \dots, e_n\}$. An embedding model f_θ maps text to vectors, and cosine similarity ranks the evidence:

$$h_c = f_\theta(c) \quad h_i = f_\theta(e_i)$$

$$s(c, e_i) = (h_c \cdot h_i) / (\|h_c\| \|h_i\|)$$

$$E_k(c) = \text{TopK over } e_i \in E \text{ of } s(c, e_i)$$



- The retrieval component ranks records by similarity and returns the top k candidates.
- Retrieved passages are candidates for subsequent verification; similarity measures representational proximity, not factual correctness or fraud probability.
- A vector index such as FAISS may be used to accelerate retrieval.

Example: Claim: "The loan was repaid in full by June." → Retrieved evidence (s = high): "Final settlement of the loan was recorded in June." → forwarded to verification as a candidate, not as proof. (Illustrative.)

B. Mutation Detection

Mutation detection examines whether related claims differ in ways that may affect their meaning. Given claims c_a and c_b , the system computes semantic similarity and compares extracted attributes. The indicators considered are summarized in Table 1.

Table 1. Mutation indicators and their score terms

Indicator	What Is Compared	Illustrative Example	Score Term
Entity substitution	Named people, organizations, accounts	"Paid to Acme Ltd" → "Paid to Apex Ltd"	D_{entity}
Numerical change	Amounts, percentages	"50,000" → "5,00,000"	D_{numeric}
Temporal change	Dates, time intervals	"12 March" → "21 March"	D_{temporal}
Polarity change	Insertion or removal of negation	"was approved" → "was not approved"	D_{polarity}
Causal or relational change	Cause–effect and relationship statements	"A caused B" → "B caused A"	Semantic term
Meaning-altering paraphrase	Reworded text with altered facts	Paraphrase that shifts the stated outcome	$1 - s(c_a, c_b)$

A conceptual mutation score is:

$$M(c_a, c_b) = \alpha(1 - s(c_a, c_b)) + \beta D_{\text{entity}} + \gamma D_{\text{numeric}} + \delta D_{\text{temporal}} + \varepsilon D_{\text{polarity}}$$

Here, s is semantic similarity, while the D terms represent entity, numerical, temporal, and polarity differences. The coefficients α , β , γ , δ , and ε determine their relative contributions. They must be calibrated using validation data or justified domain-specific rules; the equation describes a proposed design, not an empirically validated model. A high mutation score indicates a potentially meaningful difference that warrants review. It does not establish malicious intent, because legitimate updates and corrections can produce similar signals.

C. Entity Extraction and Graph Construction

The graph is represented as $G = (V, R)$, where V is the set of nodes and R is the set of relationships. Graph construction consists of entity extraction, conservative entity resolution, node creation, relationship identification, provenance attachment, and computation of graph features. Relationships retain provenance, timestamps, types, and confidence information where available.



Table 2. Graph node types

Node Type	Description	Typical Attributes
Claim	A textual statement under assessment	Claim ID, text, extracted attributes
Evidence Document	A retrieved record or passage	Document ID, excerpt, source, timestamp
Person / Organization	Named parties mentioned in records	Resolved name, aliases, identifiers
Account	Financial or service account	Account ID, owner link, status
Transaction	A recorded transfer or payment	Amount, date, parties
Event	A dated occurrence relevant to a claim	Event type, date, location

Table 3. Graph relationship types

Relationship	Meaning	Typical Direction
SUPPORTED_BY	Evidence supports the claim	Claim → Evidence
CONTRADICTED_BY	Evidence contradicts the claim	Claim → Evidence
MENTIONS_ENTITY	Record refers to an entity	Record → Entity
RELATED_TO	General semantic or contextual relation	Node → Node
POTENTIAL_MUTATION_OF	Claim is a possibly altered version of another	Claim → Claim
SHARES_ATTRIBUTE_WITH	Nodes share an identifier or attribute	Node ↔ Node
OCCURRED_BEFORE	Temporal ordering of events or records	Event → Event

Possible graph features include node degree, shared-neighbor counts, connected components, temporal relationships, and weighted paths. Graph neural networks may be evaluated if sufficient labelled data is available. A transparent graph using explicit relationship rules is a valid starting point and may be easier to audit than a complex neural architecture.

D. Evidence-Aware Verification

The verification module evaluates a claim using retrieved evidence and the relevant graph neighborhood. For claim c , evidence $E_k(c)$, and graph neighborhood G_c , the system estimates:

$$P(y \mid c, E_k(c), G_c), \quad y \in \{\text{Supported, Refuted, Insufficient Evidence}\}$$

- Retrieve relevant passages and identify supporting and contradicting evidence.
- Compare entities and numerical attributes between claim and evidence.
- Incorporate relevant graph context.
- Produce a label with traceable evidence.
- Optionally use a language model for claim extraction, evidence summarization, or explanation generation; generated explanations must remain grounded in retrieved evidence.

Importantly, evidence that a statement is false does not necessarily establish that the statement was made fraudulently. Fraud determination requires additional context and appropriate investigation.

E. Confidence Assessment and Abstention

The framework is not forced to produce a definitive answer when evidence is inadequate. Let $p_{\max}$ be the highest calibrated class probability and τ a decision threshold:

$$\text{Prediction} = \text{highest-probability class, if } p_{\max} \geq \tau$$

$$\text{Prediction} = \text{Insufficient Evidence, if } p_{\max} < \tau$$



This is a conceptual rule; deployment requires calibration on held-out data and safeguards for conflicting evidence, poor retrieval, and out-of-distribution cases. Abstention may also occur in the following situations, and the system records why it abstained:

- No relevant evidence is retrieved.
- Sources conflict.
- Entities cannot be resolved reliably.
- Inputs fall outside the supported domain.

F. Evidence Traceability

Each verification output includes the fields listed in Table 4, enabling reviewers to reconstruct the analytical path behind a decision.

Table 4. Fields of a traceable verification output

Field	Description
Claim identifier	Unique ID of the assessed claim
Predicted verification label	Supported, Refuted, or Insufficient Evidence
Calibrated confidence	Calibrated probability of the label, where available
Evidence identifiers and excerpts	IDs and relevant passages used in the decision
Supporting / contradicting classification	Role of each evidence item relative to the claim
Graph nodes and relationships	Relevant graph context for the claim
Mutation indicators	Detected entity, numeric, temporal, or polarity changes
Abstention status and reason	Whether the system abstained and why
Model version and timestamp	Models used and processing time

G. Fraud Intelligence Output

The framework distinguishes three analytical levels (Table 5). The automated system primarily supports the first two; formal decisions remain subject to appropriate human review.

Table 5. Analytical levels of fraud intelligence output

Level	Name	Description	Performed By
Level 1	Evidence Assessment	Determines whether evidence supports, refutes, or fails to establish a claim	Automated system
Level 2	Suspicious-Pattern Assessment	Identifies potentially meaningful changes, anomalies, or relationships that merit investigation	Automated system
Level 3	Human-Reviewed Fraud Determination	Considers evidence, policies, and investigation findings before reaching a formal conclusion	Human reviewers

5. Scoring and Evaluation Formulas

Core formulas of the framework and the standard measures used for evaluation:

- Semantic Similarity: $s(c, e_i) = (h_c \cdot h_i) / (|h_c| |h_i|)$
- Mutation Score: $M(c_a, c_b) = \alpha(1 - s) + \beta D_{\text{entity}} + \gamma D_{\text{numeric}} + \delta D_{\text{temporal}} + \varepsilon D_{\text{polarity}}$
- Abstention Rule: Insufficient Evidence if $p_{\max} < \tau$, otherwise the highest-probability class
- Recall@k = Queries with relevant evidence in top k / Total queries
- Mean Reciprocal Rank: $MRR = (1 / |Q|) \sum 1 / \text{rank}_i$
- Coverage = Cases answered / Total cases
- Selective Risk = Errors among answered cases / Cases answered
- Expected Calibration Error: $ECE = \sum_b (|B_b| / n) \cdot |\text{accuracy}(B_b) - \text{confidence}(B_b)|$



IV. SYSTEM ARCHITECTURE OVERVIEW

The proposed information flow is: input claims and records → preprocessing → semantic embedding → evidence retrieval → mutation analysis → graph construction → evidence-aware verification → confidence assessment → traceable output. Every module preserves record provenance and exposes structured inputs and outputs, so that components can be replaced or evaluated independently.

The architecture is organized in layers covering input handling, retrieval, analysis, verification, persistence, service integration, and the investigation interface. Table 6 lists the suggested components for each layer. These are implementation options; the final paper must list the actual technologies, versions, models, and configurations used.

Table 6. Suggested software components by layer

Layer	Component	Technology
Input Processing	Normalization, segmentation, provenance assignment	Python and NLP libraries
Semantic Embedding	Claim and evidence vectorization	Sentence-embedding model
Retrieval	Top-k similarity search	FAISS or equivalent vector index
Entity Extraction	Named entities and deterministic attributes	NER and deterministic extractors
Graph Analysis	Relationship storage and graph features	NetworkX or graph database
Verification	Evidence-grounded claim labelling	NLI classifier or evidence-grounded language model
API Integration	Service endpoints for the pipeline	FastAPI or equivalent
Persistence	Records, evidence metadata, audit data	Relational database and metadata storage
Evaluation	Metrics and reproducible experiments	scikit-learn and experiment scripts
Interface	Investigation and review workspace	Web-based investigation dashboard

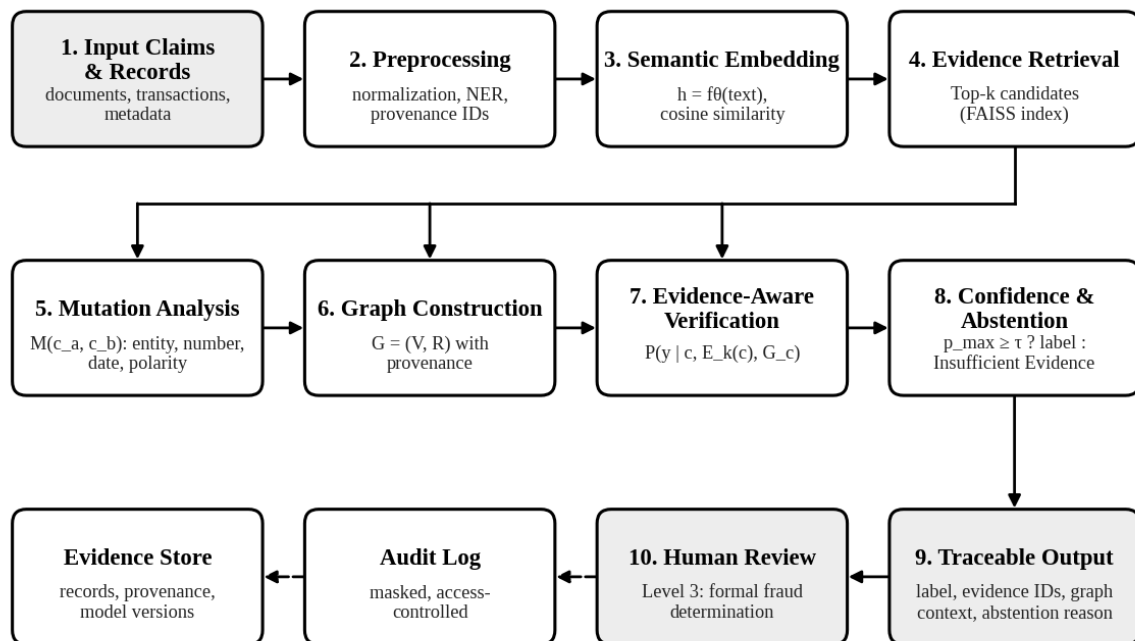


Fig. 1: System Architecture



1. Model Management

Each model is associated with its version, configuration, supported languages, and evaluation results. Model changes are tracked because different embedding and language models can produce different outputs. The system handles rate limits, network failures, invalid responses, and service unavailability without generating unsupported conclusions.

2. Security and Privacy

Fraud intelligence may involve sensitive personal or financial information. The implementation applies the following safeguards:

- Data minimization, access controls, secure credential management, and appropriate encryption.
- Masking of sensitive fields where possible.
- Audit logs that avoid unnecessary storage of confidential content.
- Treatment of retrieved documents and user-supplied text as untrusted data; instructions embedded within documents must not override system policies or control tools.

V. EXPERIMENTAL DESIGN

1. Research Hypotheses

The hypotheses in Table 7 must be tested rather than assumed to be true.

Table 7. Research hypotheses

ID	Hypothesis	Related RQ
H1	Semantic retrieval achieves better evidence recall than a lexical baseline.	RQ1
H2	Mutation-aware features improve identification of material claim changes compared with semantic similarity alone.	RQ2
H3	Graph context provides additional value for relationship-dependent cases.	RQ3
H4	Calibrated abstention reduces unsupported high-confidence decisions, with a measurable coverage trade-off.	RQ4
H5	Evidence-linked outputs improve traceability during structured human review.	RQ4, RQ5

2. Datasets

Table 8. Datasets for evaluation

Dataset	Purpose	Evidence Type	Notes
FEVER	Supported / Refuted / NotEnoughInfo claim verification	Unstructured text	185,445 claims with evidence annotations [1]
FEVEROUS	Verification with mixed evidence	Text, tables, and lists	Extends FEVER to structured evidence [2]
Fraud-specific dataset	Direct evaluation of financial fraud detection	Institution-specific records	Public or institutionally authorized; to be identified

Generic fact-verification labels do not directly represent financial fraud. The evaluation must not report fact-verification accuracy as fraud-detection accuracy.

3. Data Preparation

- Divide the dataset into training, validation, and test partitions where appropriate.
- Prevent information leakage: paraphrases of the same original claim, or records derived from the same source, must not be distributed across partitions in ways that make evaluation artificially easy.
- Document preprocessing decisions, label mappings, duplicate removal, missing-value handling, and class distributions.
- Keep test data separate from model tuning and threshold selection.



4. Baseline Systems

Baselines use comparable data splits and evaluation protocols.

Table 9. Baseline systems

ID	Baseline	Description
B1	Lexical	TF-IDF with cosine similarity, or a simple classifier
B2	Semantic	Dense retrieval without graph or mutation analysis
B3	Verification	Text-based verifier with retrieved evidence
B4	Mutation	Semantic similarity plus attribute differences
B5	Graph	Relationship-based features
B6	Integrated framework	All proposed components combined

5. Evaluation Metrics

Table 10. Evaluation metrics by task

Task	Metrics
Evidence retrieval	Recall@k, Precision@k, Mean Reciprocal Rank
Claim verification	Accuracy; macro-averaged precision, recall, and F1-score; per-class precision and recall; confusion matrix
Mutation detection	Precision, recall, F1-score; false-positive rate for legitimate changes; performance by mutation type
Abstention and calibration	Coverage, selective risk, expected calibration error, risk-versus-coverage curves
Operational performance	Average and percentile inference latency, throughput, memory usage, index size and storage overhead

6. Ablation Study

The contribution of individual modules is evaluated by removing them from the full system (Table 11). Changes in retrieval quality, verification performance, calibration, and coverage will determine whether each component provides measurable value.

Table 11. Ablation configurations

Configuration	Component Removed	Question Addressed
Full system	None	Reference performance
Without mutation analysis	Mutation detection	Does mutation analysis add value (H2)?
Without graph context	Graph construction and features	Does graph context help (H3)?
Without evidence-aware verification	Verification module	Is evidence-conditioned labelling needed?
Without abstention	Confidence threshold rule	What coverage and risk change (H4)?
Without dense semantic retrieval	Dense retrieval	Does semantic retrieval beat lexical (H1)?



VI. RESULTS AND DISCUSSION

No quantitative results are asserted in this paper because verified experimental measurements have not been supplied. The tables below define the reporting protocol and must be completed after experiments are executed. Claims of improvement should include metric values, datasets, test conditions, and confidence intervals or significance analysis where justified.

1. Evidence Retrieval Performance

Table 12. Evidence retrieval performance

Method	Recall@5	Recall@10	MRR
TF-IDF baseline	TBD	TBD	TBD
Dense semantic retrieval	TBD	TBD	TBD
Integrated retrieval pipeline	TBD	TBD	TBD

2. Claim-Verification Performance

Table 13. Claim-verification performance

Method	Accuracy	Macro-F1	Evidence Score
Text-only baseline	TBD	TBD	TBD
Evidence-conditioned verifier	TBD	TBD	TBD
Integrated framework	TBD	TBD	TBD

3. Mutation-Detection Performance

Table 14. Mutation-detection performance

Method	Precision	Recall	F1-Score
Semantic difference only	TBD	TBD	TBD
Entity and attribute comparison	TBD	TBD	TBD
Combined mutation detector	TBD	TBD	TBD

4. Abstention and Calibration Performance

Table 15. Abstention performance

Configuration	Coverage	Error Rate at Coverage	Calibration Error
No abstention	TBD	TBD	TBD
Fixed confidence threshold	TBD	TBD	TBD
Calibrated abstention	TBD	TBD	TBD



5. Operational Performance

Table 16. Operational performance

Metric	Measurement	Result
Inference latency	Average and percentile (P50, P95, P99)	TBD
Throughput	Records processed per second	TBD
Memory usage	Peak memory during retrieval and verification	TBD
Index size	Vector index and storage overhead	TBD
Graph construction overhead	Time and memory per record batch	TBD

6. Ablation Results

Table 17. Ablation results

Configuration	Recall@k	Macro-F1	Coverage	Calibration Error
Full system	TBD	TBD	TBD	TBD
Without mutation analysis	TBD	TBD	TBD	TBD
Without graph context	TBD	TBD	TBD	TBD
Without evidence-aware verification	TBD	TBD	TBD	TBD
Without abstention	TBD	TBD	TBD	TBD
Without dense retrieval	TBD	TBD	TBD	TBD

7. Interpretation Plan by Research Question

- **RQ1:** Compare retrieval metrics to determine whether semantic embeddings recover evidence missed by lexical matching.
- **RQ2:** Examine whether mutation features identify meaningful changes while limiting false positives from harmless paraphrases.
- **RQ3:** Compare text-only and graph-augmented verification on relationship-dependent cases.
- **RQ4:** Examine the relationship between coverage and error rate, and report the trade-off transparently.
- **RQ5:** Measure the operational overhead of graph construction, evidence verification, and model inference.

8. Discussion

a. Evidence-Centered Intelligence

The framework separates evidence retrieval from verification: retrieval proposes candidate material, while verification evaluates whether the material supports a claim. Evidence identifiers and provenance can make assessments more auditable than isolated classification scores. However, traceability does not guarantee that evidence is correct, current, or complete.

b. Handling Evolving Patterns

Mutation analysis aims to distinguish meaningful changes from legitimate variation. Entity substitutions, numerical changes, temporal changes, and altered polarity should be evaluated separately. An unusual change is not automatically suspicious; context and evidence must be considered.

c. The Role of Graph Context

Graphs provide a structured view of connections across records and can help investigators identify shared attributes, recurring entities, and chains of related evidence. However, graph quality depends on entity resolution and relationship extraction. Incorrectly merging entities can create misleading connections, so graph outputs retain provenance and support manual inspection.



d. Uncertainty and Human Oversight

Abstention is important when incorrect conclusions could have serious consequences, and a system that always produces a definitive answer may conceal uncertainty. The framework allows uncertain cases to be escalated for further evidence collection or human review. Its effectiveness must be measured using selective risk, coverage, calibration, and review quality.

9. Limitations and Error Analysis

The framework has the limitations listed in Table 18, together with the safeguards built into the design.

Table 18. Limitations and design safeguards

Limitation	Description	Design Safeguard
Evidence quality	Missing, outdated, or unreliable evidence can produce incorrect results	Provenance and timestamps; abstention on missing evidence
Domain transfer	Generic fact-verification performance may not generalize to financial fraud	Fraud-specific evaluation; separate reporting
Semantic ambiguity	Embedding models may retrieve related but non-entailing passages	Retrieval used for discovery only; verification step
Mutation false positives	Legitimate corrections may resemble suspicious mutations	Score is a review signal, not a verdict
Graph construction errors	Incorrect entity resolution may create misleading connections	Conservative resolution; provenance on edges
Model calibration	Confidence may be unreliable across domains and shifting data	Held-out calibration; distribution-shift tests
Adversarial behavior	Manipulated text and obfuscation may reduce reliability	Untrusted-input handling; human review
Computational cost	Dense retrieval, language models, and graphs raise latency and infrastructure needs	Operational metrics; modular components
Privacy and fairness	Sensitive records require access control and assessment of disparate error rates	Data minimization, masking, audit logs
Human dependence	Evidence-linked output supports but does not replace accountable decisions	Level 3 human-reviewed determination

10. Comparison with Conventional Approaches

Table 19 summarizes the design intent of the proposed framework relative to conventional approaches. These are design characteristics, not measured results.

Table 19. Conventional approaches versus the proposed framework

Feature	Conventional Systems	Abhimanyu Chakravyuha (Proposed Design)
Matching of claims	Exact keywords or fixed rules	Semantic retrieval with embeddings
Changed narratives	Evaded by small edits	Mutation analysis of entities, numbers, dates, polarity
Relationship analysis	Records examined in isolation	Graph of claims, entities, and evidence
Explainability	Score without evidence	Evidence-linked, traceable output
Uncertainty	Forced definitive output	Abstention with recorded reason
Decision authority	Often automated flag	Human-reviewed fraud determination
Validation status	Established practice	Requires experimental validation



11. Practical Applications

Potential applications of the framework include:

- Prioritizing suspicious records for investigators.
- Identifying related claims across records.
- Detecting changes in recurring narratives.
- Linking evidence to entities.
- Reviewing contradictory statements.
- Producing traceable investigation summaries.

These are proposed applications rather than validated deployments or demonstrated financial-loss prevention results.

VII. CONCLUSION

1. Concluding Remarks

This paper presented Abhimanyu Chakravyuha, a proposed evidence-aware AI framework for fraud intelligence integrating semantic retrieval, mutation detection, graph-based relationship analysis, and claim verification. Its principal objective is to connect analytical assessments to identifiable evidence while explicitly accounting for uncertainty. The framework distinguishes semantic similarity from factual support, suspicious patterns from confirmed fraud, and automated analysis from human-reviewed decisions.

The proposed methodology establishes a foundation for systematic evaluation. Nevertheless, the effectiveness of the framework cannot be established by architecture alone. Reproducible experiments, suitable baselines, domain-specific datasets, ablation studies, and transparent reporting are necessary to determine whether the proposed components deliver measurable benefits. Abhimanyu Chakravyuha should therefore be understood as an evidence-centered research framework whose practical value depends on rigorous validation, responsible deployment, and continuous evaluation.

2. Future Work

- Domain-specific evaluation using appropriately labelled fraud records and expert-reviewed cases.
- Testing mutation detection against annotated examples of material and non-material changes covering entities, quantities, dates, polarity, and context.
- Extending graph reasoning with temporal graphs or graph neural networks where sufficient data is available, compared against transparent graph-feature baselines.
- Evaluating multilingual verification with language-specific datasets rather than assuming multilingual embeddings alone are sufficient.
- Testing confidence calibration and abstention under distribution shifts.
- Human-in-the-loop studies measuring whether evidence-linked outputs help analysts review cases more efficiently and avoid unsupported conclusions.

REFERENCES

Fact Extraction and Verification Datasets

- [1]. J. Thorne, A. Vlachos, C. Christodoulopoulos, and A. Mittal, "FEVER: A Large-scale Dataset for Fact Extraction and VERification," Proc. NAACL-HLT, 2018, pp. 809–819. DOI: 10.18653/v1/N18-1074. [Online]. Available: <https://aclanthology.org/N18-1074/>
- [2]. R. Aly et al., "The Fact Extraction and VERification Over Unstructured and Structured information (FEVEROUS) Shared Task," Proc. Fourth Workshop on Fact Extraction and VERification, 2021. [Online]. Available: <https://aclanthology.org/2021.fever-1.1/>

Graph-Based Analysis and Misinformation Detection

- [3]. H. T. Phan, N. T. Nguyen, and D. Hwang, "Fake news detection: A survey of graph neural network methods," Applied Soft Computing, vol. 139, art. 110235, 2023. DOI: 10.1016/j.asoc.2023.110235.
- [4]. S. Gong, R. O. Sinnott, J. Qi, and C. Paris, "Fake News Detection Through Graph-based Neural Networks: A Survey," arXiv:2307.12639, 2023. [Online]. Available: <https://arxiv.org/abs/2307.12639>

Knowledge Graphs for Fact Checking

- [5]. U. Qudus, M. Röder, M. Saleem, and A.-C. Ngonga Ngomo, "Fact Checking Knowledge Graphs—A Survey," ACM Computing Surveys, vol. 58, no. 1, art. 8, 2025. DOI: 10.1145/3749838.