

A Unified System for Intelligent Financial Fraud Detection

Vojislav Stojkovic
Morgan State University
vojislav.stojkovic@morgan.edu

Author Note

Correspondence concerning this article should be addressed to Vojislav Stojkovic, Morgan State University.

<https://doi.org/10.68414/DFQT4711>

Abstract

Modern financial ecosystems require real-time, transparent, and trustworthy fraud detection systems capable of processing massive transaction volumes while meeting regulatory and auditing requirements. This paper presents a unified, modular system integrating four layers: (a) a blockchain-based transaction logging mechanism, (b) an explainable artificial intelligence (XAI) fraud detection engine, (c) a Mistral large language model (LLM) client for natural-language reasoning, and (d) an AI agent for orchestration. A Python-based prototype demonstrates the complete pipeline—from transaction ingestion to fraud scoring, explanation generation, and immutable blockchain storage. The system serves as an educational and experimental platform for research and development, offering flexibility and extensibility for future enhancements.

Keywords: financial fraud detection, explainable AI, Mistral large language models, AI agents.

Introduction

Financial institutions increasingly rely on intelligent systems to detect suspicious behavior, mitigate risk, and comply with regulatory standards. As transaction volumes grow and fraud patterns evolve, traditional rules-based systems often fail to provide the adaptability, transparency, and auditability required in modern financial environments.

The FinTech Transaction Simulation component provides a sandbox environment for evaluating fraud rules, validating payment flows, conducting risk analysis, and training AI/ML

models using synthetic data. Fraud, broadly defined as intentional deception for financial gain, manifests in numerous forms, including identity theft, credit card fraud, insurance fraud, investment fraud, and online payment fraud. Its consequences, financial loss, reputational damage, and legal penalties - underscore the need for robust detection systems.

The unified system presented in this study addresses four essential questions:

1. Is the transaction fraudulent?
2. Why was it flagged or approved?
3. How can the decision be explained in natural language?
4. Can the decision be stored immutably for auditing?

To answer these questions, the system integrates four major layers: Blockchain, Explainable AI (XAI) Fraud Detection, Mistral LLM Client, and AI Agent. Together, these layers form a modular, transparent, and extensible architecture for intelligent fraud detection.

Literature Review

AI Agents and Multi-Agent System Design

Recent research highlights a shift from monolithic AI models toward autonomous and semi-autonomous agent-based architectures. Albada (2025) describes practical approaches for building AI-driven applications using modular agents capable of reasoning and tool use. Lakshmanan and Hapke (2025) identify generative AI design patterns that support orchestration, memory, and tool integration. Dibia (2025) provides a systematic framework for multi-agent system design, emphasizing communication protocols and scalability. Collectively, these works demonstrate the growing relevance of agentic systems in complex domains such as fraud detection.

AI Agents for Fraud Detection and Reasoning

Agent-based reasoning systems have gained traction in fraud detection research. Bai et al. (2025) introduce AuditAgent, a multi-agent framework for cross-document fraud evidence discovery. Chen et al. (2026) propose SAGE, a self-reflective LLM-driven agentic system capable of iterative reasoning over evolving fraud patterns. These studies highlight the advantages of adaptive, reasoning-driven agents over static rule-based systems.

Explainable AI in Finance

Explainability is essential in financial AI due to regulatory constraints and stakeholder trust requirements. Černevičienė and Kabašinskas (2024) provide a systematic review of XAI methods in finance, emphasizing transparency and interpretability. Jain et al. (2025) argue that hybrid models combining machine learning with post-hoc explainability are increasingly favored. Kadambala (2024) demonstrates how explainable ML enhances risk assessment in blockchain-based payment systems. Mostafa (2026) emphasizes regulatory compliance and governance in U.S. banking fraud detection.

Multi-Source and Blockchain-Based Fraud Detection

Blockchain-based fraud detection systems have emerged as a promising direction for ensuring data integrity and traceability. Nti et al. (2025) propose a multi-source XAI framework integrating supervised and unsupervised learning to detect blockchain transaction manipulation, demonstrating the value of heterogeneous data sources.

Methodology

This study employs a multi-layered experimental system design methodology. The system integrates blockchain simulation, explainable AI fraud detection, AI agent orchestration, and LLM-based natural language reasoning. Each layer is implemented as an independent module within a flexible architecture.

Blockchain Layer

The blockchain layer provides immutable storage of fraud decisions and explanations. Each block contains transaction data, an explanation, and a cryptographic hash linking it to the previous block. Any modification invalidates the chain, ensuring integrity, traceability, transparency, accountability, compliance, and nonrepudiation.

The following Python code is an implementation of a simple blockchain.

BLOCKCHAIN

class Block: # Represents a single block in the blockchain.

def __init__(self, index, timestamp, transaction, explanation, previous_hash):

self.index = index

self.timestamp = timestamp

self.transaction = transaction

self.explanation = explanation

```

        self.previous_hash = previous_hash
        self.hash = self.calculate_hash()
    def calculate_hash(self):
        block_string = json.dumps({
            "index": self.index,
            "timestamp": self.timestamp,
            "transaction": self.transaction,
            "explanation": self.explanation,
            "previous_hash": self.previous_hash
        }, sort_keys=True)
        return hashlib.sha256(block_string.encode()).hexdigest()
class Blockchain: # Manages the chain of blocks.
    def __init__(self):
        self.chain = [self.create_genesis_block()]
    def create_genesis_block(self):
        return Block(0, time.time(), {"type": "GENESIS"}, "Genesis Block", "0")
    def latest_block(self):
        return self.chain[-1]
    def add_block(self, transaction, explanation):
        previous = self.latest_block()
        block = Block(len(self.chain), time.time(), transaction, explanation, previous.hash)
        self.chain.append(block)

```

Explainable AI (XAI) Fraud Detection Layer

The XAI layer evaluates transactions using rule based scoring and produces human interpretable reasoning. Signals include transaction amount, frequency, device characteristics, and geographic indicators. Decisions fall into three categories: Approve, Review, or Reject.

The following Python code is an implementation of simple XAI Fraud Detector.

XAI FRAUD DETECTION

```

class FraudDetector:
    def evaluate(self, tx):
        score = 0

```

```

reasons = []
if tx["amount"] > 5000: score += 40; reasons.append("Transaction amount exceeds $5000")
if tx["frequency"] > 10: score += 30; reasons.append("Unusually high transaction frequency")
if tx["new_device"]: score += 20; reasons.append("Transaction initiated from new device")
if tx["foreign_country"]: score += 20; reasons.append("Foreign country transaction")
if score >= 70: decision = "REJECT"
elif score >= 40: decision = "REVIEW"
else: decision = "APPROVE"
return decision, score, reasons

```

AI Agent Layer

The AI Agent Layer coordinates the workflow by integrating the detector, explainer, and blockchain. It performs task decomposition, routing, context sharing, and governance.

The following python code is an implementation of simple AI agent FraudAgent.

AI AGENT

class FraudAgent:

def __init__(self, detector, explainer, blockchain):

```

self.detector = detector
self.explainer = explainer
self.blockchain = blockchain

```

def process_transaction(self, tx):

```

decision, score, reasons = self.detector.evaluate(tx)
explanation = self.explainer.explain(tx, decision, score, reasons)
self.blockchain.add_block(tx, explanation)
return {"decision": decision, "score": score, "reasons": reasons, "explanation": explanation}

```

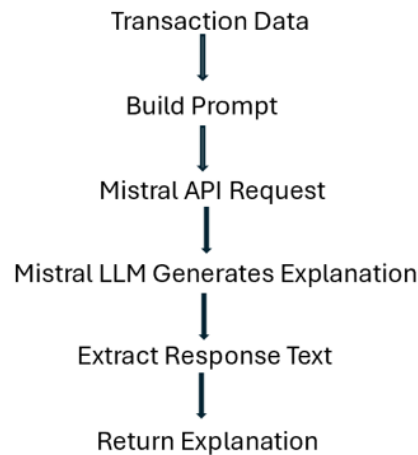
Mistral Large Language Models (LLM) Client Layer

The Mistral LLM Client generates natural language explanations based on transaction data, risk scores, and fraud indicators. It produces an executive summary, risk analysis, and recommendation under 200 words.

Figure 1 presents a pipeline diagram describing how the Mistral LLM Client processes some transaction data and turns it into a human-readable explanation.

Figure 1

Pipeline Diagram



The following python code is an implementation of simple Mistral LLM Client. The code defines a python class `MistralExplainer` that sends fraud-analysis information to the Mistral AI API and returns a natural-language explanation. `MistralExplainer` is responsible for generating fraud explanations using Mistral LLM.

MISTRAL LLM CLIENT

class MistralExplainer:

```
def __init__(self, api_key, model="mistral-small-latest"):
```

```
    self.api_key = api_key
```

```
    self.model = model
```

```
def explain(self, transaction, decision, score, reasons):
```

```
    prompt = f"""
```

```
    You are a financial fraud analyst.
```

```
    Transaction: {json.dumps(transaction, indent=2)}
```

```
    Decision: {decision}
```

```
    Risk Score: {score}
```

```
    Risk Factors: {reasons}
```

```
    Generate: 1. Executive Summary; 2. Risk Analysis; 3. Recommendation
```

```
    Keep explanation under 200 words."""
```

```
    url = "https://api.mistral.ai/v1/chat/completions"
```

```
    # Mistral chat-completion endpoint headers:
```

```
    # Authenticate the request using the API key
```

```
    # Tell the server the request body contains JSON
```

```

headers = {"Authorization": f"Bearer {self.api_key}", "Content-Type": "application/json"}
payload = {"model": self.model, "messages": [{"role": "user", "content": prompt}]}
# Sending the Request
response = requests.post(url, headers=headers, json=payload)
# Parse the Response
result = response.json()
# Return the Generated Explanation
return result["choices"][0]["message"]["content"]

```

main() Function

This main() function is the entry point of the Intelligent Financial Fraud Detection program. It creates the required components, processes a sample transaction, displays the results, and shows the blockchain trail.

The following python code is the main() function.

MAIN

def main():

```

API_KEY = os.getenv("MISTRAL_API_KEY")
# Creates a blockchain object that will store transaction records in blocks.
blockchain = Blockchain()
# Creates an AI-powered explanation generator.
explainer = MistralExplainer(api_key=API_KEY)
# Creates the fraud detection engine.
detector = FraudDetector()
# Create the Fraud Agent
agent = FraudAgent(detector, explainer, blockchain)
# Create a sample transaction
transaction = {
    "transaction_id": "MD1001",
    "customer_id": "VS12345",
    "amount": 10000,
    "frequency": 25,
    "new_device": True,
    "foreign_country": True
}

```

```

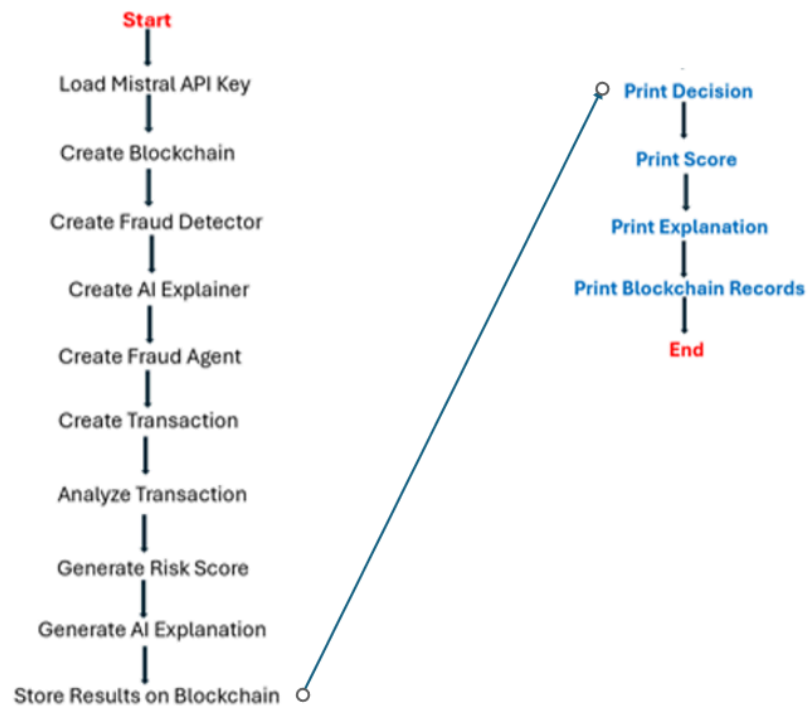
# Process the transaction
result = agent.process_transaction(transaction)
print("\nDECISION"); print("=" * 50); print(result["decision"])
print("\nRISK SCORE"); print("=" * 50); print(result["score"])
print("\nEXPLANATION"); print("=" * 50); print(result["explanation"])
print("\nBLOCKCHAIN"); print("=" * 50)
for block in blockchain.chain:
    print(f"\nBlock #{block.index}");
    print("Hash:", block.hash[:20], "...")
    print("Previous:", block.previous_hash[:20], "...")

```

Figure 2 presents a system workflow diagram showing how a fraud-analysis + AI-explanation pipeline runs from start to finish.

Figure 2

System Workflow Diagram



Transaction

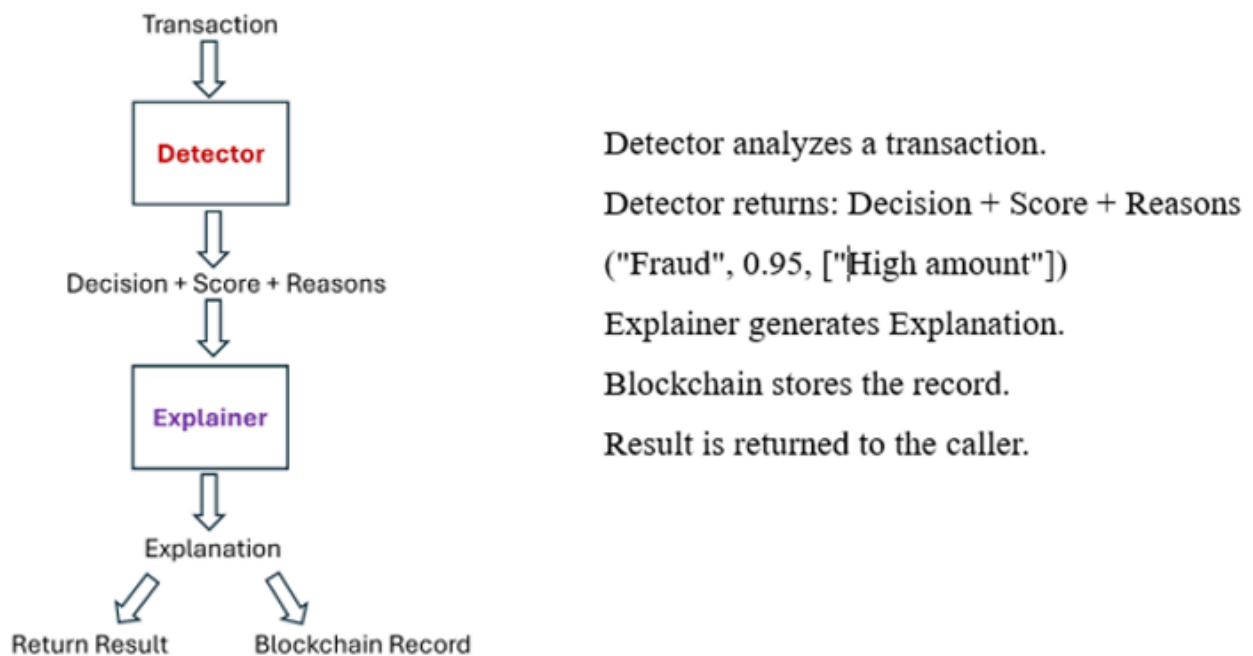
Transaction consists of Detector and Explainer.

Figure 3 presents a component-level architecture diagram for a fraud-analysis system. It shows

how a transaction flows through two major modules - a Detector and an Explainer - and then how the results are returned and optionally written to a blockchain.

Figure 3

Component-Level Architecture Diagram



Test Example

The Unified System for Intelligent Financial Fraud Detection has been tested only through a single sample transaction. The results demonstrate that the system runs. The results do not provide quantitative evaluation using a dataset, baseline comparisons, or metrics such as precision, recall, F1-score, AUC, or false positive/false negative analysis. The LLM explanation component is also not evaluated for faithfulness, reliability, or hallucination risk.

Python 3.13.9 | packaged by Anaconda, Inc. | (main, Oct 21 2025, 19:09:58) [MSC v.1929 64 bit (AMD64)]

Type "copyright", "credits" or "license" for more information.

IPython 9.7.0 -- An enhanced Interactive Python. Type '?' for help.

%runfile C:/Users/vojis/FraudDetection.py --wdir

DECISION

REJECT

RISK SCORE

110

EXPLANATION

Executive Summary

The transaction **MD1001** was rejected due to a **high-risk profile** (Risk Score: 110). Key concerns include an unusually large amount (\$10,000), high transaction frequency (25), a new device, and a foreign country transaction. These factors collectively indicate elevated fraud risk.

Risk Analysis

- **Amount Risk**: The transaction exceeds the \$5,000 threshold, suggesting potential money laundering or unauthorized large transfers.
- **Frequency Risk**: High transaction frequency (25) may indicate automated or suspicious activity.
- **Device Risk**: A new device increases the likelihood of unauthorized access.
- **Geographic Risk**: Foreign transactions are inherently riskier due to cross-border fraud patterns.

Recommendation

Reject the transaction and **flag the customer (VS12345)** for enhanced due diligence. Consider:

- **Temporary account freeze** pending manual review.
- **Customer verification** (e.g., identity confirmation, device authentication).
- **Transaction monitoring** for future patterns.

This decision aligns with fraud prevention best practices and mitigates financial exposure.

BLOCKCHAIN

Block #0

Hash: e8cc04143ab987144e27 ...

Previous: 0 ...

Block #1

Hash: fc55d5a17536e84b0f78 ...

Previous: e8cc04143ab987144e27 ...

Results

The results of this work are primarily system-level experimental outcomes obtained from the implemented prototype. The study demonstrates the feasibility and effectiveness of integrating blockchain simulation, explainable AI (XAI), large language models (LLMs), and AI agents into a unified fraud detection pipeline.

The developed system is a fully functional prototype that:

- Detects fraudulent activity in simulated financial transactions
- Generates both model-based and human-readable explanations
- Records decisions immutably using blockchain-inspired principles
- Coordinates workflows through AI-driven agents
- Provides modular and extensible architecture suitable for future research and enhancement

Discussion

Extensive research has been conducted on fraud detection, blockchain security, explainable AI (XAI), large language models (LLMs), and intelligent agents as independent domains. However, comparatively limited work has explored their integration into a unified system architecture.

This study contributes to this emerging area by combining transaction simulation, explainable fraud detection, natural language explanation generation via an LLM, blockchain-based immutable record storage, and agent-based orchestration within a modular framework. This integrated design enhances transparency, auditability, and adaptability, addressing several key challenges in modern financial ecosystems, particularly those related to trust, interpretability, and system coordination.

This paper includes both the Python source code and corresponding test cases. As a result, it reads more like a technical prototype report or an educational system demonstration than a full research article. The inclusion of the code is valuable because it allows readers to examine the implementation details directly. Without the code, the paper would feel overly general and lack the necessary technical depth.

Limitations and Future Work

The main limitation of this work is that the fraud detection component is primarily rule-based. Rule-based systems are fast and explainable, but they cannot keep up with evolving fraud patterns, suffer from high false positives, and require constant manual tuning. Improving the work means augmenting or transforming the rule-based system into a layered, adaptive, data-driven architecture. The most effective improvement is to integrate machine learning (ML), anomaly detection, and dynamic rule orchestration into a layered fraud-detection pipeline. This reduces false positives, adapts to new fraud patterns, and scales better than static rules.

Future enhancements to the system will include:

- Machine learning improvements: Integration of more advanced models, deep neural networks, or reinforcement learning-based fraud detection
- Local LLM deployment: Use of models such as Mistral via frameworks like Ollama or vLLM for improved privacy and efficiency
- Blockchain expansion: Integration with real blockchain platforms
- Fabric for more realistic distributed ledger deployment
- User interface development: Design of interactive dashboards for monitoring fraud detection outcomes and exploring blockchain records

Conclusion

This work presents a unified system for financial transaction processing that integrates blockchain technology, explainable AI-based fraud detection, a Mistral-based large language model interface, and AI agent coordination.

The proposed system demonstrates how modern financial infrastructures can incorporate transparency, interpretability, and security into intelligent fraud detection pipelines. It enables not only the identification of fraudulent transactions but also the generation of clear explanations and the secure, immutable recording of decisions for auditing and compliance purposes.

References

- Albada, M. (2025). *Building applications with AI agents*. O'Reilly Media.
- Bai, S., Wu, B., Zhang, Y., Wu, C., Zheng, X., Yuan, Y., Wu, K., & Li, J. (2025). AuditAgent: Expert-guided multi-agent reasoning for cross-document fraudulent evidence discovery. *Proceedings of the 6th ACM International Conference on AI (ICAIF '25)*, 1–9.
<https://doi.org/10.1145/3768292.3770383> ([doi.org in Bing](#))
- Černevičienė, J., & Kabašinskas, A. (2024). Explainable artificial intelligence (XAI) in finance: A systematic literature review. *Artificial Intelligence Review*, 57, Article 216.
<https://link.springer.com/article/10.1007/s10462-024-10854-8> ([link.springer.com in Bing](#))
- Chen, Y., Li, S., Liang, Y., Wang, L., & Liu, R. (2026). SAGE: An LLM-driven self-reflective agentic framework for fraud detection. *arXiv preprint*, arXiv:2606.08146.
- Dibia, V. (2025). *Designing multi-agent systems: Principles, patterns, and implementation for AI agents*. <https://github.com/victordibia/designing-multiagent-systems> ([github.com in Bing](#))
- Jain, A., Kulkarni, R., & Lin, S. (2025). Explainable AI in big data fraud detection.
https://www.researchgate.net/publication/398851517_Explainable_AI_in_Big_Data_Fraud_Detection ([researchgate.net in Bing](#))
- Kadambala, K. M. (2024). Explainable machine learning models for risk assessment in blockchain payment gateways. *International Journal of AI, Big Data, Computational and Management Studies*, 5(2), 161–172.
<https://doi.org/10.63282/3050-9416.IJAIBDCMS-V5I2P116> ([doi.org in Bing](#))
- Lakshmanan, V., & Hapke, H. (2025). *Generative AI design patterns: Solutions to common challenges when building GenAI agents and applications*. O'Reilly Media.
- Mostafa, M. (2026). Explainable AI in U.S. banking fraud detection: A comparative framework for regulatory compliance, model transparency, and risk governance.
<https://ssrn.com/abstract=6129286> ([ssrn.com in Bing](#))
- Nti, I. K., Nyarko-Boateng, O., Boateng, S., Weyori, B. A., & Adekoya, A. F. (2025). Multi-source explainable AI for blockchain transaction manipulation detection using supervised and unsupervised models.
<https://www.researchgate.net/publication/392792643> ([researchgate.net in Bing](#))