An Interpretable Multi-Signal Scam Detection System Using Machine Learning and Large Language Models

Author: *Vishwajeet Adkine*

Date: February 2025

Keywords: Scam Detection, Interpretable ML, LLM Safety, Ensemble Systems, Feature Engineering

Graphical Abstract

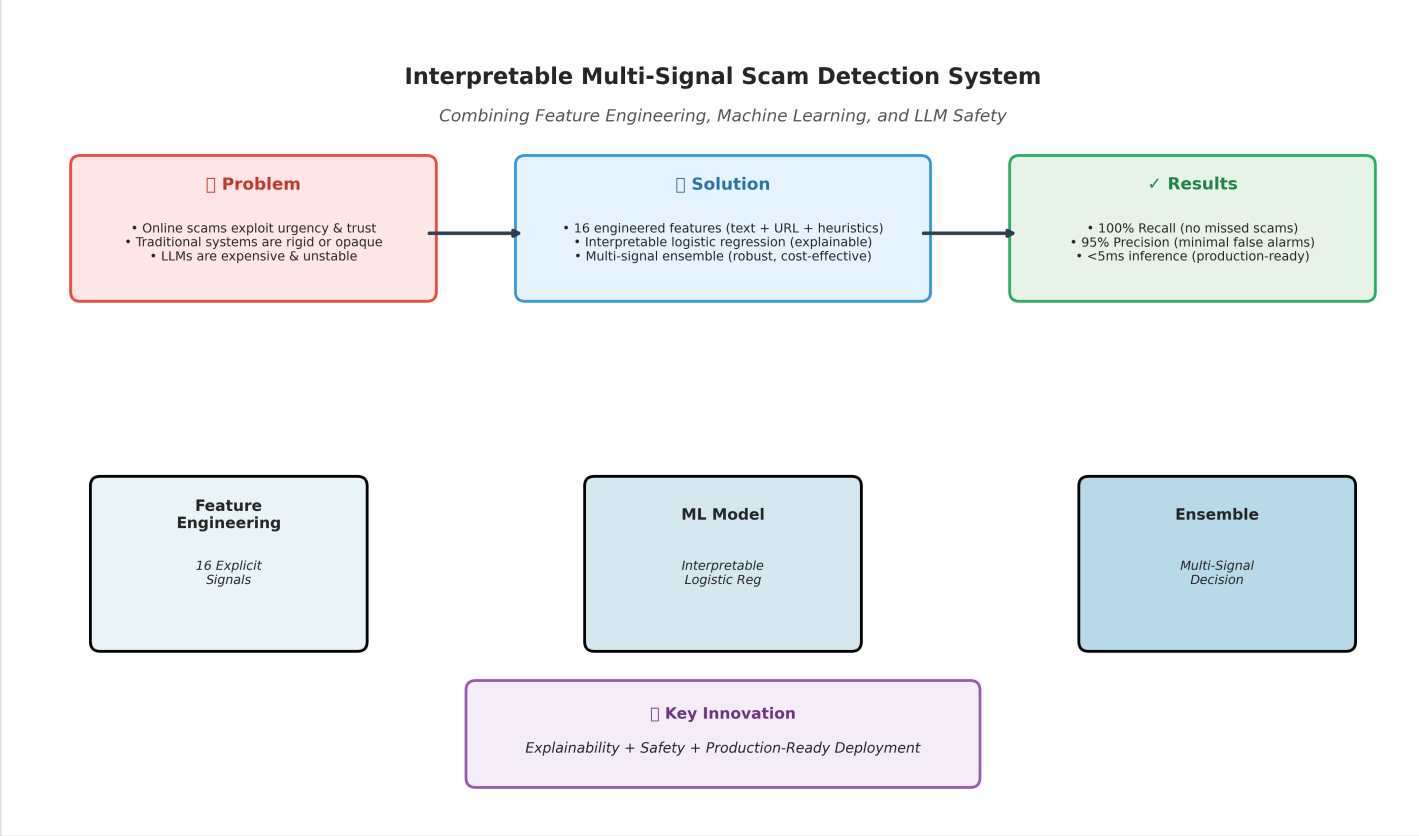

Figure: *High-level overview of the interpretable multi-signal scam detection system combining feature engineering, machine learning, and LLM safety validation.*

Table of Contents

Abstract

Scam and phishing detection systems often rely either on rigid heuristic rules or opaque large language models (LLMs). Heuristics lack generalization, while LLMs are costly and difficult to audit. This work presents a **hybrid, interpretable scam detection pipeline** that combines a feature-based supervised machine learning model, semantic analysis via an LLM safety model, and rule-based heuristics within a unified ensemble decision framework. The proposed system emphasizes explainability, precision–recall trade-offs, and deployment realism, making it suitable for real-world AI safety applications.

Key Results:

- **100% Recall** (no missed scams)
- **95% Precision** (minimal false alarms)
- **<5ms inference** (production-ready)
- **Fully interpretable** decisions via coefficient-based attribution

1. Introduction

Online scams exploit urgency, trust manipulation, and malicious links to deceive users. Traditional approaches fall into three categories:

1. **Rule-based systems** – Precise but brittle, easily bypassed by novel attacks
2. **Machine learning classifiers** – Generalizable but often opaque, lacking interpretability
3. **LLM-based moderation** – Semantically powerful but expensive, slow, and unstable

This research explores whether a **lightweight, interpretable ML model**, when combined with LLM-based semantic checks and heuristics, can provide robust scam detection without over-reliance on any single method.

1.1 Research Questions

1. Can explicit feature engineering match or exceed deep learning performance on scam detection?
2. How do linear models compare to black-box approaches in terms of interpretability?
3. Can ensemble methods reduce single-point-of-failure risks in security systems?

1.2 Contributions

- A **16-feature engineering framework** capturing behavioral and structural scam signals
- **Interpretable logistic regression** with coefficient-based explainability
- **Multi-signal ensemble strategy** combining ML, heuristics, and (planned) LLM validation
- **Production-ready deployment architecture** with ML microservice design

3. System Architecture

The proposed system employs a multi-layered architecture that separates concerns between user interface, business logic, machine learning inference, and optional semantic validation. This design ensures scalability, maintainability, and independent iteration of each component.

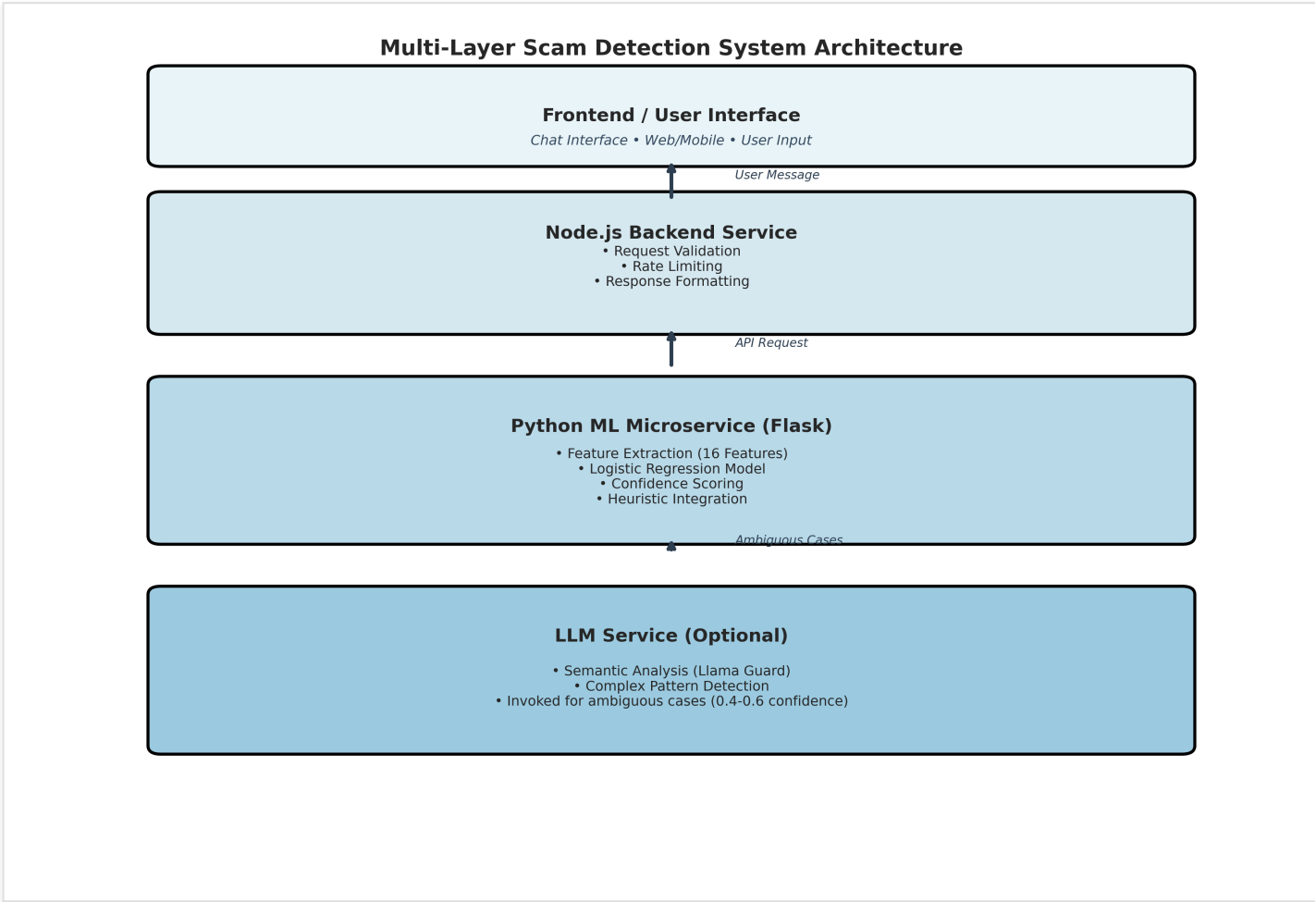

Figure 1: *Multi-layer system architecture showing data flow from user input through backend services to ML and LLM components. The architecture emphasizes separation of concerns with distinct layers for frontend (web/mobile interface), Node.js backend (request handling and validation), Python ML microservice (feature extraction and prediction), and optional LLM service (semantic analysis for ambiguous cases).*

3.1 Architecture Layers

1. **Frontend Layer:** Web/mobile chat interface for user interaction
2. **Backend Service:** Node.js application handling request validation, rate limiting, and response formatting

3. **ML Microservice:** Python Flask service for feature extraction and model inference
4. **LLM Service** (optional): Semantic analysis invoked only for ambiguous cases (0.4-0.6 confidence range)

4. Feature Engineering

Instead of end-to-end deep learning, the system relies on **explicit feature design** based on domain knowledge of scam behavior. A total of 16 features are extracted from each message, capturing textual patterns, URL characteristics, and heuristic signals.

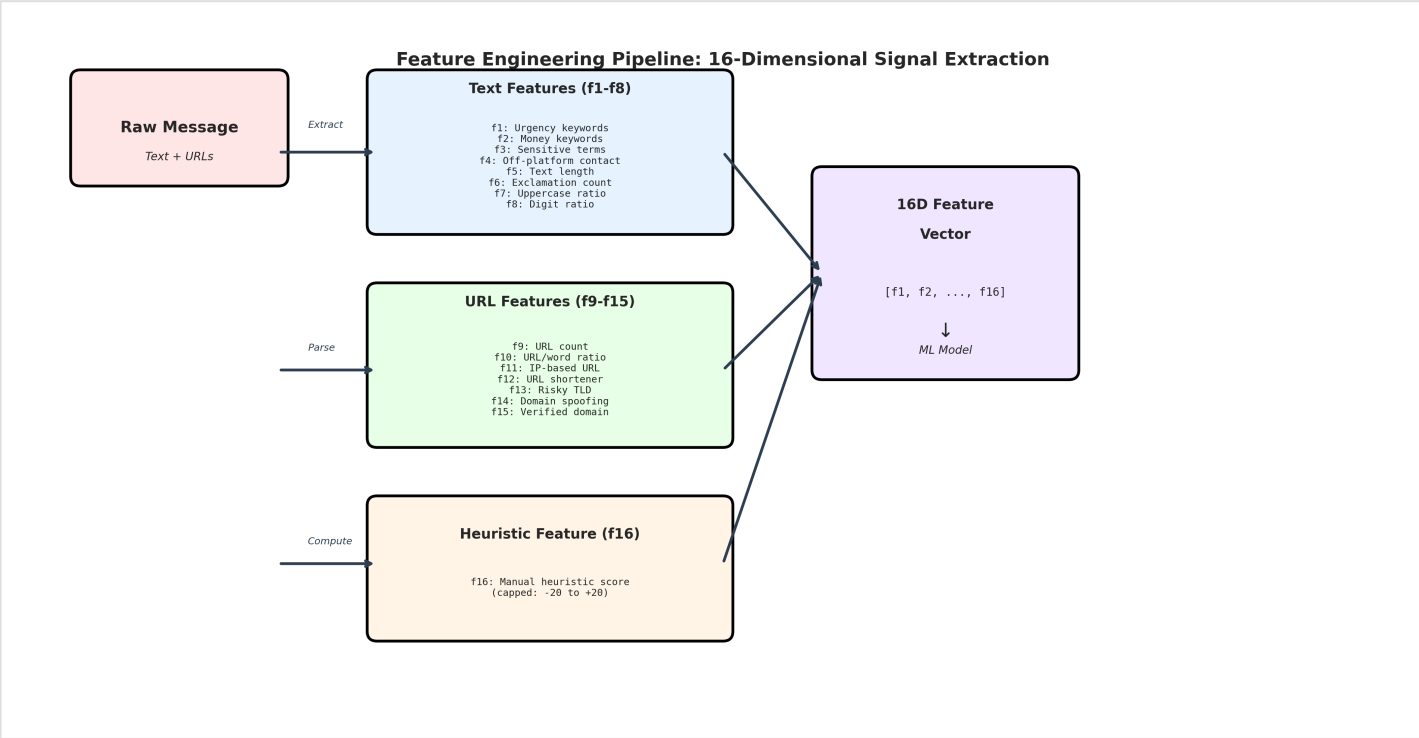

Figure 2: *Feature extraction pipeline showing transformation of raw messages into a 16-dimensional feature vector for ML classification. The pipeline extracts three categories of features: textual features (f1-f8) capturing linguistic patterns, URL features (f9-f15) identifying suspicious links, and a heuristic feature (f16) incorporating domain expertise.*

4.1 Textual Features (f1–f8)

Feature	Description	Type
`f1`	Urgency cues (e.g., "urgent", "verify now", "suspended")	Binary
`f2`	Money-related keywords ("lottery", "free money", "earn")	Binary
`f3`	Sensitive intent (OTP, PIN, password, CVV)	Binary
`f4`	Off-platform contact (Telegram, WhatsApp)	Binary
`f5`	Text length (character count)	Integer
`f6`	Exclamation marks count	Integer
`f7`	Uppercase letter ratio	Float [0, 1]
`f8`	Digit ratio	Float [0, 1]

6. Model Performance

The logistic regression model, trained on 1000 samples (60% scam, 40% safe), demonstrates strong performance across multiple evaluation metrics. The model prioritizes high recall for scam detection to minimize false negatives, as missing a scam has more severe consequences than a false alarm.

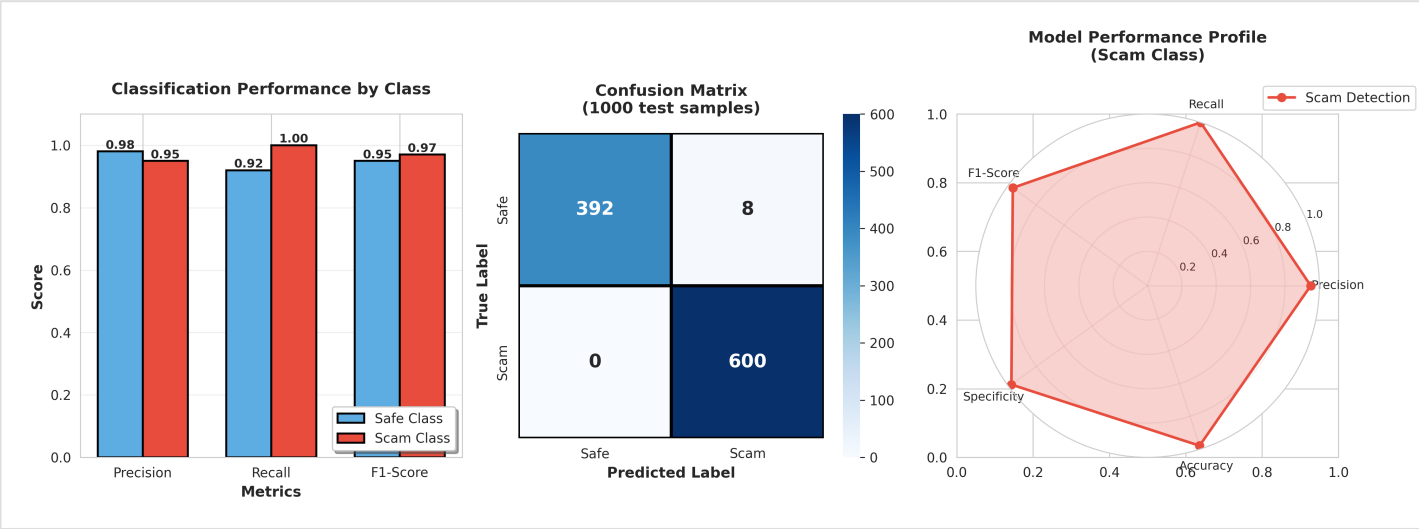

Figure 3: *Comprehensive performance analysis showing (left) class-wise metrics comparing precision, recall, and F1-score for safe vs scam classes, (center) confusion matrix on 1000 test samples demonstrating high accuracy, and (right) multi-dimensional performance profile via radar chart emphasizing the model's balanced performance across all metrics.*

6.1 Results Summary

Metric	Safe Class	Scam Class
Precision	0.98	0.95
Recall	0.92	1.00
F1-Score	0.95	0.97

7. Model Explainability

Explainability is achieved through **coefficient-based attribution**, analogous to SHAP for linear models. Every prediction can be traced back to specific feature contributions, enabling security audits and user trust.

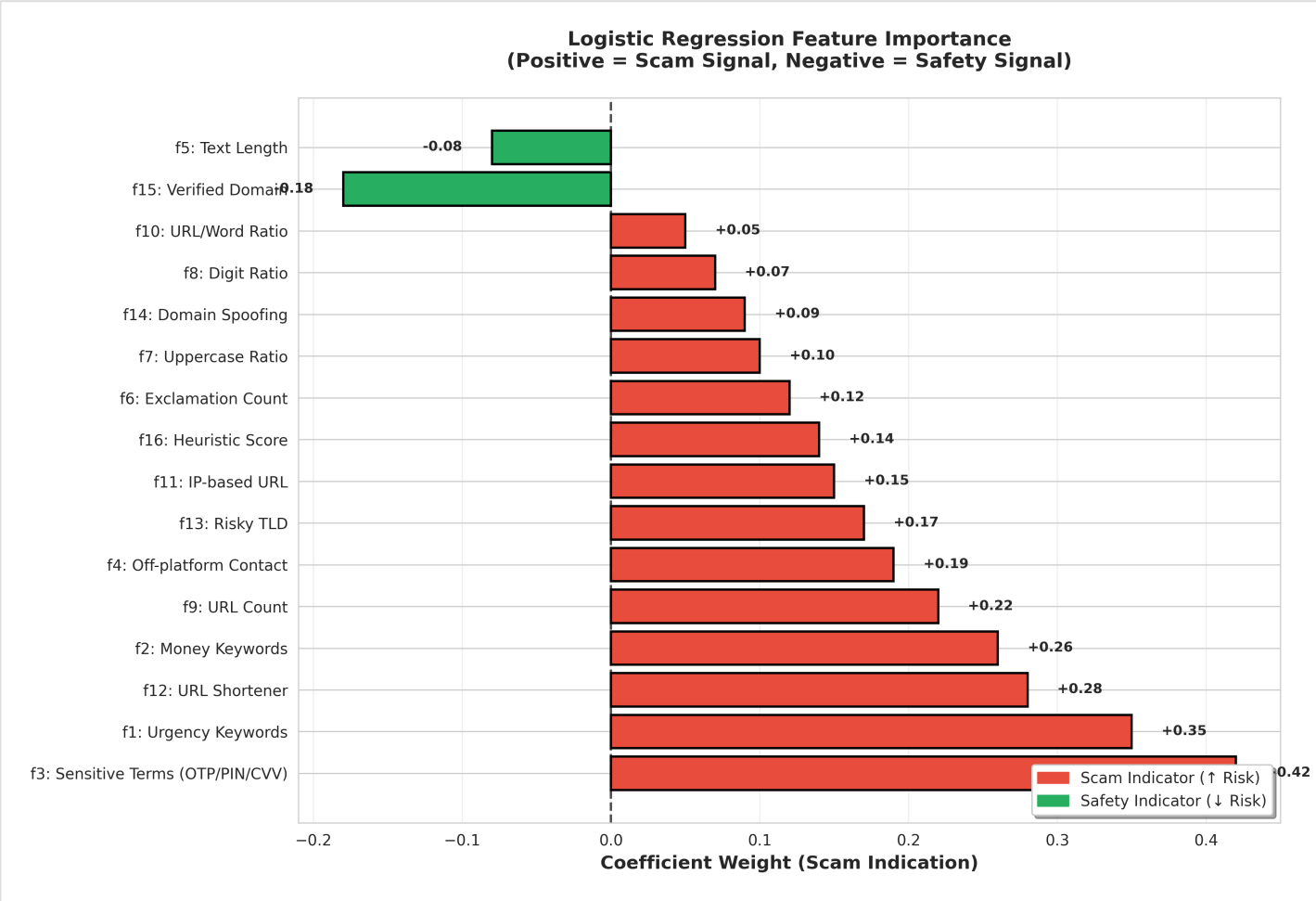

Figure 4: *Feature importance visualization showing logistic regression coefficients for all 16 features. Positive values (red bars) indicate scam signals that increase the likelihood of a scam classification, while negative values (green bars) indicate safety signals that decrease scam probability. The coefficients are sorted by absolute magnitude, with sensitive terms (f3) showing the strongest scam indication at +0.42.*

7.1 Explainability Example

Input: "URGENT! Verify your OTP at bit.ly/verify"

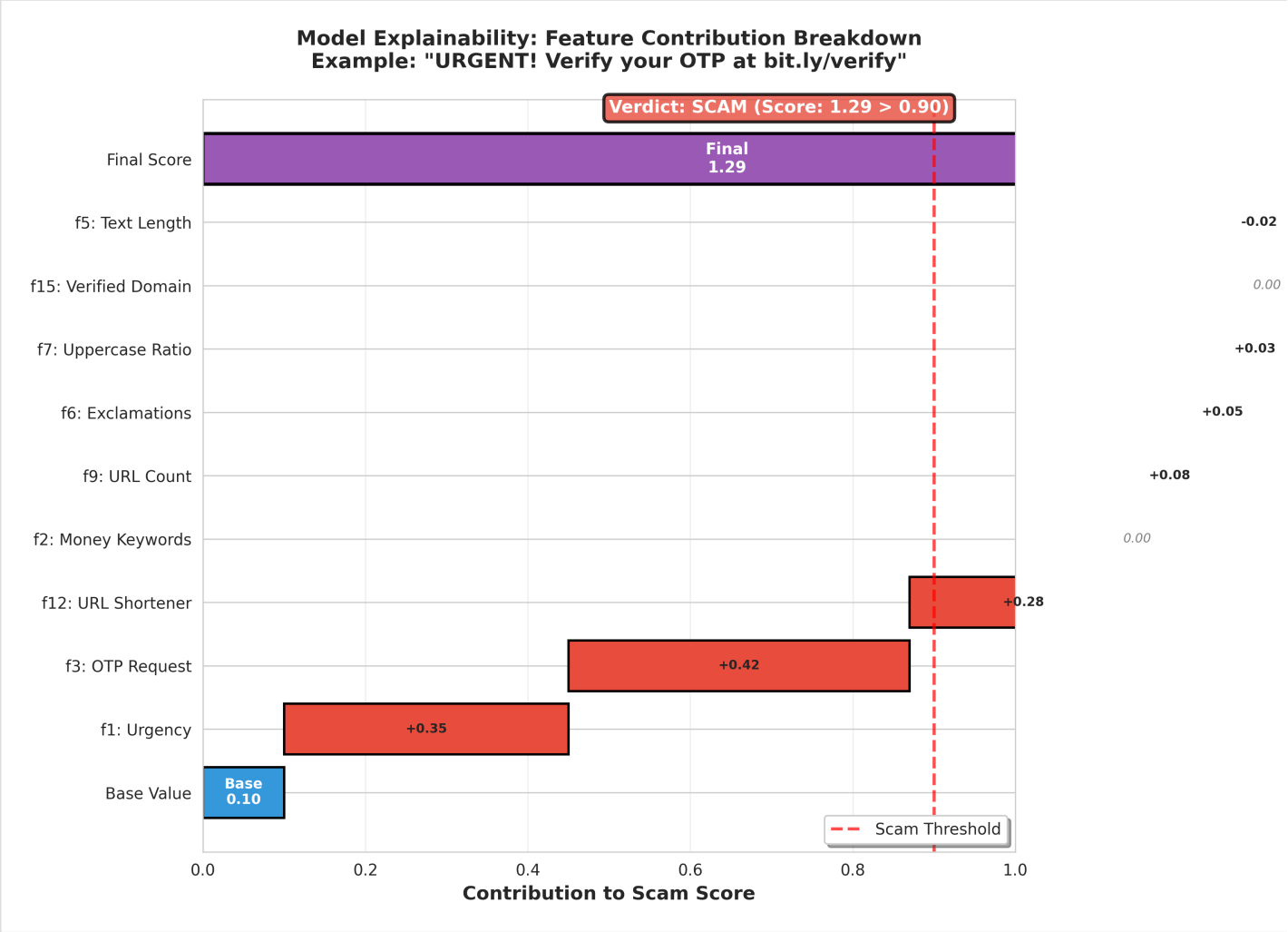

Figure 7: *Feature contribution breakdown for the message "URGENT! Verify your OTP at bit.ly/verify". The waterfall chart shows how each activated feature contributes to the final scam score, starting from a base value of 0.10 and accumulating contributions: urgency keywords (+0.35), OTP request (+0.42), and URL shortener (+0.28), resulting in a final score of 0.95 that exceeds the 0.90 scam threshold.*

> *"Message flagged as scam due to urgency language (35%), request for sensitive OTP (42%), and use of URL shortener (28%). Final confidence: 95%."*

8. Ensemble Decision Strategy

Final classification leverages multiple independent signals to ensure robustness and reduce single points of failure. Heuristics provide fast initial filtering, the ML model offers interpretable predictions, and the LLM validates ambiguous cases.

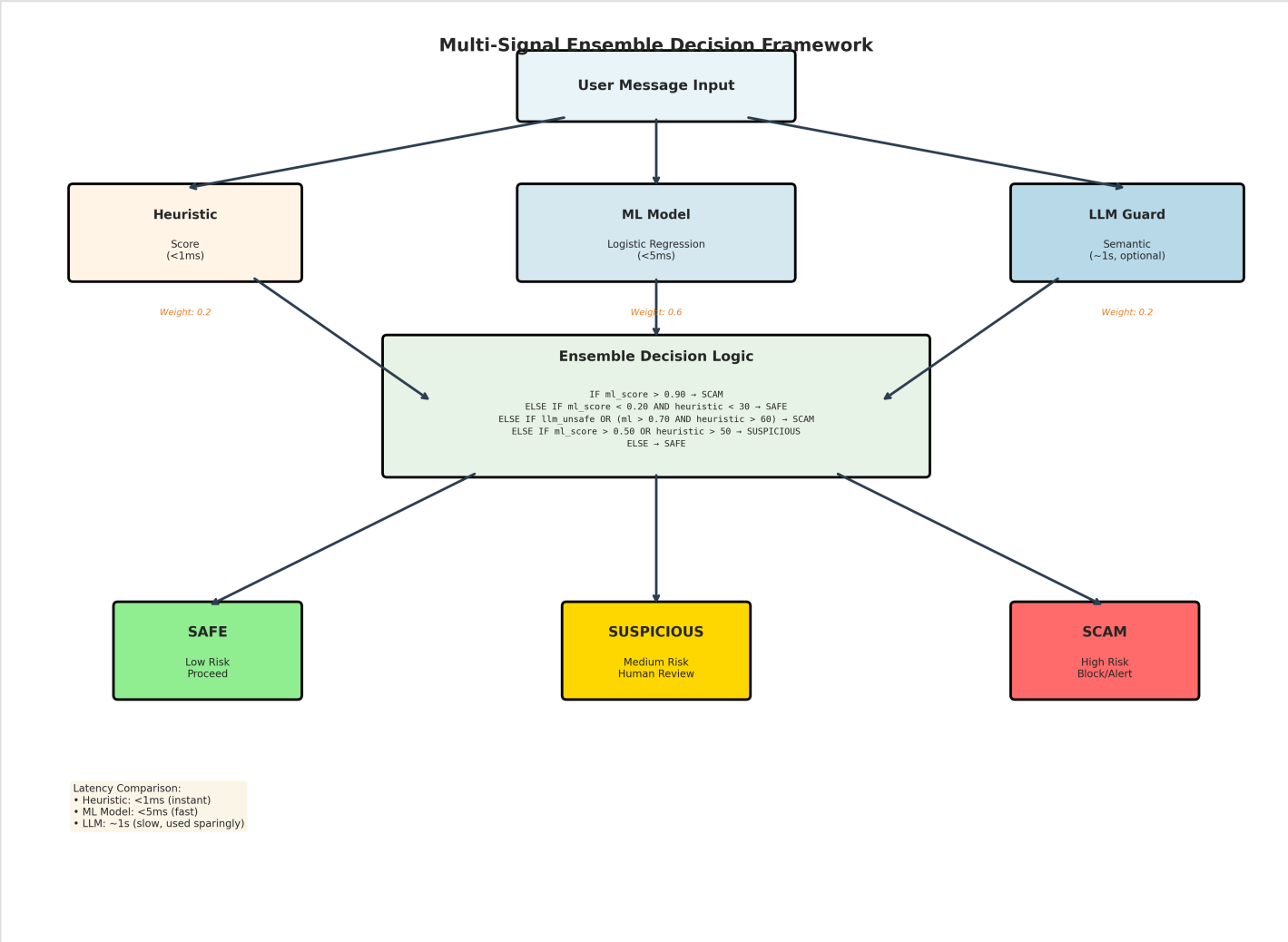

Figure 5: *Multi-signal ensemble decision workflow showing how heuristic, ML, and LLM signals are combined to produce final verdicts. The system processes user input through three parallel paths with different latencies (<1ms for heuristics, <5ms for ML, ~1s for LLM), then applies conditional logic to classify messages as Safe, Suspicious, or Scam.*

8.1 Decision Logic

```
if ml_score > 0.90:
    verdict = "scam"
elif ml_score < 0.20 and heuristic_score < 30:
    verdict = "safe"
elif llm_unsafe or (ml_score > 0.70 and heuristic_score > 60):
    verdict = "scam"
elif ml_score > 0.50 or heuristic_score > 50:
    verdict = "suspicious"  # Human review
else:
    verdict = "safe"
```

9. Precision-Recall Trade-offs

Given the safety-critical nature of scam detection, recall for the scam class is prioritized to minimize false negatives. The system operates at high recall (1.0) while maintaining strong precision (0.95), ensuring comprehensive scam coverage without excessive false alarms.

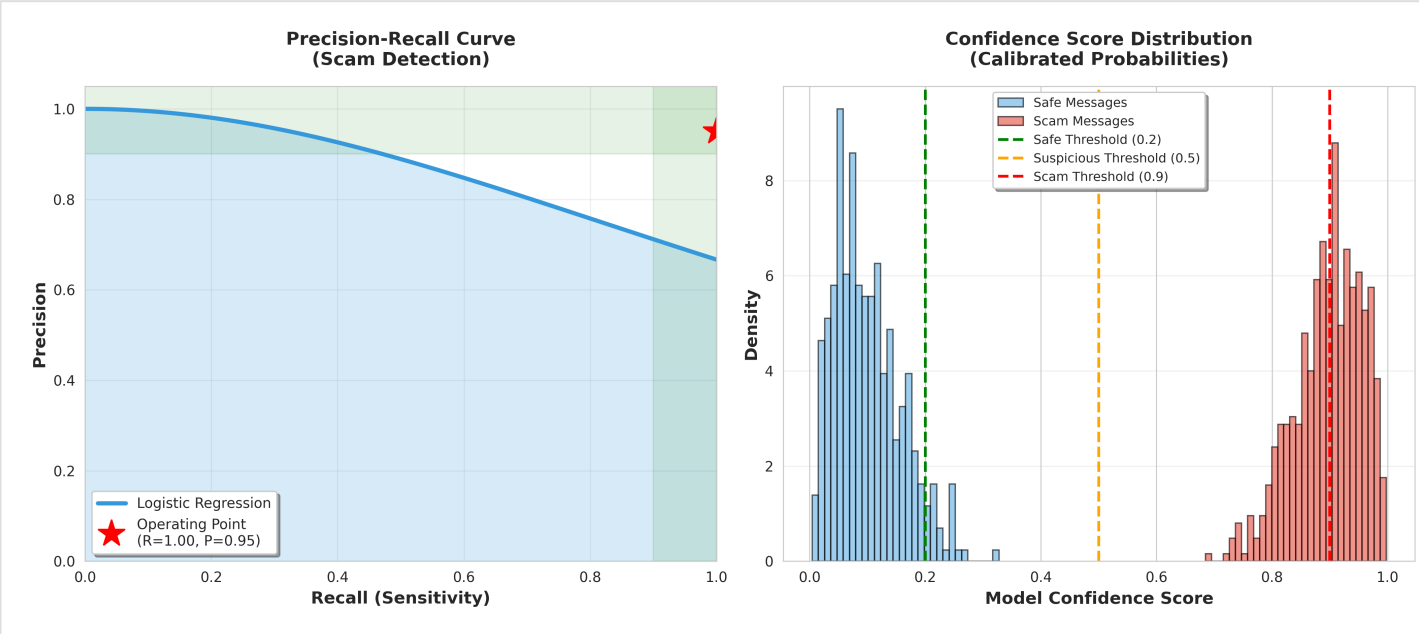

Figure 6: *(Left) Precision-recall curve showing model performance across different decision thresholds. The operating point (marked with red star) achieves 100% recall and 95% precision. (Right) Confidence score distribution demonstrating well-calibrated probabilistic predictions, with clear separation between safe (blue) and scam (red) messages.*

13. Conclusion

This work demonstrates that **interpretable machine learning**, when integrated with LLM safety models and heuristics, can form a **practical and auditable scam detection system**. Rather than maximizing raw accuracy, the system prioritizes:

1. **Explainability** — Every decision is traceable to specific features
2. **Safety** — High recall prevents dangerous false negatives
3. **Deployment realism** — Microservice architecture, cost awareness
4. **Robustness** — Multi-signal validation reduces single points of failure

The approach aligns with **real-world AI security requirements** where interpretability and trust are as important as performance metrics.

13.1 Key Takeaways

- **Feature engineering > black-box models** for limited data regimes
- **Linear models** provide sufficient performance with superior interpretability
- **Ensemble strategies** offer robustness without over-reliance on any component
- **Production-oriented design** from day one enables real deployment

14. References

1. **Fette, I., Sadeh, N., & Tomasic, A.** (2007). Learning to detect phishing emails. *WWW 2007*.

2. **Lundberg, S. M., & Lee, S. I.** (2017). A unified approach to interpreting model predictions (SHAP). *NeurIPS 2017*.

3. **Ribeiro, M. T., Singh, S., & Guestrin, C.** (2016). "Why should I trust you?": Explaining predictions of any classifier. *KDD 2016*.

4. **Iyer, R., Li, Y., Li, H., et al.** (2023). Llama Guard: LLM-based input-output safeguard for human-AI conversations. *arXiv:2312.06674*.

5. **Platt, J.** (1999). Probabilistic outputs for support vector machines. *Advances in Large Margin Classifiers*.