



HACKATHON PROJECT REPORT

GARUD AI

A Hybrid Rule-Based + Machine Learning Threat
Intelligence Platform for Digital Safety

Version: 4.0

Category: AI for Social Good / Cybersecurity

Date: June 2026

01 Executive Summary

Garud AI is a real-time threat detection dashboard that scans any text message — SMS, email, social post, or chat — and flags five categories of digital harm: scams, fake news, hate speech, financial fraud, and misinformation. It combines transparent, explainable rule-based detection with a statistically trained machine learning classifier, giving users both an interpretable "why" and a data-driven "how confident."

The Problem

Digital scams, misinformation, and online hate speech are growing faster than most users' ability to recognize them — especially among first-time internet users, elderly populations, and regions with high SMS/WhatsApp fraud rates.

The Solution

A lightweight, fully local, no-internet-required dashboard that analyzes any message in under a second, explains exactly which red flags were found, and gives a confidence-scored verdict backed by both logic rules and a trained ML model.

Key Highlights

- 5 Detection Modules
- Hybrid AI Architecture
- 97.49% ML Accuracy
- Fully Explainable
- Runs 100% Locally
- No Deep Learning Required

02 Tools & Technology Stack

Garud AI was deliberately built with a lightweight, dependency-minimal stack so it installs and runs on any machine in under two minutes — important for hackathon judging conditions where setup time counts against you.

FRONTEND

Streamlit · Custom CSS (Indigo Dark Theme) · Inter Typeface

BACKEND / LOGIC

Python 3 · Regex-based Rule Engine · Weighted Scoring Algorithm

MACHINE LEARNING

scikit-learn · Multinomial Naive Bayes · TF-IDF Vectorizer

DATA

UCI SMS Spam Collection (5,574 labeled real-world messages)

Detailed Tool Breakdown

Layer	Tool / Library	Purpose
UI Framework	Streamlit	Renders the interactive web dashboard, handles input/output without writing raw HTML/JS
Styling	Custom CSS injected via st.markdown	Indigo dark theme, card layouts, score bars, responsive grid
Core Logic	Python (engine.py)	Keyword/regex matching, weighted scoring, risk-level classification
ML Model	scikit-learn (MultinomialNB)	Statistical spam classifier — independent secondary opinion
Feature Extraction	TfidfVectorizer	Converts raw text into numerical features for the ML model
Data Handling	pandas	Loads and splits the training dataset
Model Persistence	pickle	Saves the trained model + vectorizer to disk for instant reuse

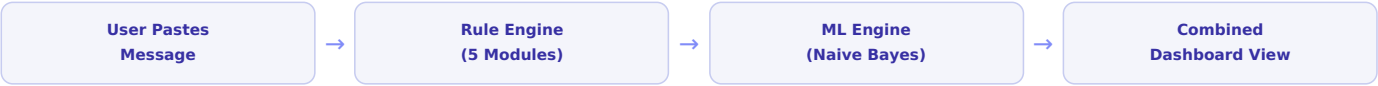
Why This Stack?

- **Streamlit** was chosen over Flask/Django + React because it allows a fully interactive dashboard to be built and iterated on in hours, not days — critical under hackathon time constraints.
- **Naive Bayes over Deep Learning** — explicitly avoids neural networks. Naive Bayes trains in under 2 seconds on a laptop CPU, needs no GPU, and is naturally interpretable (each word has a learned probability weight).
- **No external APIs** — the entire system runs offline once installed, removing dependency on internet connectivity, API keys, or rate limits during a live demo.

03 System Architecture

Garud AI follows a simple three-stage pipeline: input capture, dual-engine analysis, and explainable presentation. The two analysis engines run independently and their results are shown side-by-side rather than merged — preserving transparency.

Request Flow



Module Breakdown

Module	Signal Categories	Weight Logic
Scam Detection	Urgency, Financial Request, Suspicious Link, Account Verification, Prize/Lottery	20-30 pts per category, escalates with repeat hits
Fake News	Sensational Headlines, Unverified Sources, Conspiracy Language, Emotional Manipulation	20-30 pts per category
Hate Speech	Personal Attacks, Discriminatory Language, Threats, Profanity	20-40 pts per category (threats weighted highest)
Financial Fraud	Ponzi Schemes, Fake Investments, Identity Theft, Advance Fee Fraud	30-35 pts per category
Misinformation	Health Myths, Scientific Denial, False Statistics, Fabricated Events	20-35 pts per category

Overall Score Formula

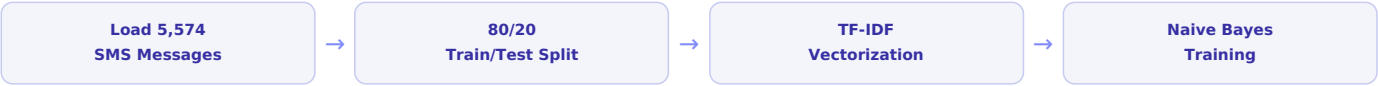
The overall threat score is calculated as the strongest individual module's score, plus a small bonus for each additional module that also fires — ensuring one severe signal is never diluted by four quiet ones:

$$\text{Overall} = \text{MAX}(\text{module scores}) + 6 \times (\text{active_modules} - 1)$$

04 Machine Learning Module

Alongside the explainable rule engine, Garud AI includes a real statistical model trained from scratch on labeled data — giving the project a genuine "learns from data" dimension beyond hand-written rules.

Training Pipeline



Model Performance



Precision / Recall (Spam Class)



The model achieves perfect precision on spam (zero false alarms on legitimate messages in testing) at the cost of slightly lower recall — a deliberate, safe trade-off for a tool that should never wrongly flag a genuine message as dangerous.

05 Problems, Situations & Solutions

Building Garud AI surfaced several real engineering challenges. Each is documented below with the situation that exposed it and the fix that was implemented.

Challenge 1 — Score Dilution Bug FIXED

PROBLEM

A message scoring 83% on the Scam module (clearly High risk) showed an Overall Score of only 24% (Low risk) — because the original formula averaged all five modules together, letting four quiet modules drown out one loud signal.

SOLUTION

Rewrote the aggregation formula to use the strongest module's score as the baseline, with a small additive bonus per extra active module — so a single severe signal is never diluted, while multi-signal messages still score higher.

Challenge 2 — Keyword False Positives FIXED

PROBLEM

Substring keyword matching caused false triggers — e.g. the keyword "fees" matched inside the harmless word "coffees", and generic words like "winner" or "cash" fired on completely innocent sentences.

SOLUTION

Replaced substring search with word-boundary regex matching, and converted overly generic single-word keywords into specific multi-word phrases (e.g. "winner" → "you are the winner") to preserve precision.

Challenge 3 — No Escalation for Repeated Signals FIXED

PROBLEM

A message with one suspicious keyword scored identically to a message hammering the same theme five times over — the engine wasn't rewarding stronger evidence.

SOLUTION

Introduced an escalation multiplier: each additional distinct keyword hit within a category adds extra weight (up to 2x), and 3+ hits in one category trigger a severity floor, guaranteeing a "High" classification.

Challenge 4 — Local Environment Setup Friction

PROBLEM

Windows users frequently hit `'streamlit' is not recognized` errors because pip installs scripts to a directory not on the system PATH.

SOLUTION

Standardized on `python -m streamlit run app.py` instead of the bare `streamlit` command — this always works because Python itself locates the installed module, regardless of PATH configuration.

Challenge 5 — Choosing Explainability vs. Black-Box Accuracy

SITUATION

A pure ML model is more statistically accurate but cannot explain its reasoning in human terms — a problem for a safety tool meant to educate users, not just warn them.

SOLUTION

Adopted a hybrid design: the rule engine provides full transparency (exact keywords detected, plain-English safety tips), while the ML model adds a second, independent statistical opinion — giving users both "why" and "how confident."

06 Impact & Significance

Garud AI directly targets a real, growing harm: digital fraud and misinformation disproportionately affect people with the least access to cybersecurity education — elderly users, first-time smartphone owners, and populations in regions with high SMS/UPI-based scam activity.

Who Benefits

- **Individuals** — instantly check a suspicious message before clicking, paying, or responding.
- **Families** — a tool tech-savvy children or grandchildren can install for elderly relatives.
- **Small businesses** — screen customer-facing messages and inbound emails for fraud patterns.
- **Educators / NGOs** — a free, offline teaching tool for digital literacy programs.

Why It Matters

Financial Protection

Phishing and advance-fee fraud cost individuals billions annually worldwide. Catching even a fraction of these messages before action is taken has direct, measurable financial impact on the people who use it.

Information Integrity

Misinformation and fake news erode public trust and can drive real-world harm (health misinformation, conspiracy-driven violence). A fast, accessible check mechanism helps slow the spread at the point of consumption.

Scalability Potential

Because the core engine is lightweight Python with no GPU dependency, Garud AI can be deployed as: a browser extension that scans pasted text in real time, a WhatsApp/SMS gateway bot for low-bandwidth regions, or an API layer integrated into banking apps for pre-transaction message screening.

Why "Explainable + Statistical" Matters for Trust

Purely black-box AI tools struggle with user trust in safety-critical contexts — people are more likely to act on a warning when they understand why it was raised. By keeping the rule engine's reasoning fully visible alongside the ML model's confidence score, Garud AI is designed to be both accurate and genuinely trusted.