SMS Spam Detection System — Product Requirements Document

Date: June 5, 2026 **Author:** Sankar Kumar Palaniappan **Program:** MIT — No Code AI and Machine Learning (Great Learning) **Status:** Final **Version:** 1.0 (All 16 evaluation criteria addressed)

---

1. Executive Summary

Problem Statement

Businesses and mobile users face an escalating threat from spam SMS messages that carry phishing links, fake prize notifications, and cyber-attack payloads. At a 13.5% spam rate across SMS traffic — confirmed in the training dataset of 5,572 messages — even a modest-volume business receives hundreds of malicious messages per day. Manual review is impossible at scale; existing keyword blocklists are trivially bypassed by spammers who continuously mutate vocabulary, abbreviations, and language patterns.

Proposed Solution

An NLP-powered SMS Spam Detection System that classifies every incoming message as ‘spam’ or ‘ham’ in real time using a Decision Tree text-classification model trained on TF-IDF features extracted from SMS content. Deployed as an API within cybersecurity infrastructure, the system intercepts spam before delivery, surfaces false-positive reports through a user feedback loop, and retrains continuously to adapt to evolving spamming techniques.

Business Impact

- **Security protection:** Prevent phishing and cyber-attack SMS from reaching users — every false negative (missed spam) is a potential security breach; the model catches 91.5% of spam on the test set
- **User trust:** Reduce spam inbox exposure, increasing confidence in SMS as a business communication channel
- **Operational efficiency:** Replace manual review and brittle keyword blocklists with automated, high-precision classification (93.50% test precision) — reducing false positives that block legitimate business messages

Key Milestones

Milestone	Target
Model development & validation (RapidMiner)	May 2024 ✅
Production API deployment	Q3 2026
Telecom / enterprise security	Q4 2026

Milestone	Target
integration	
User feedback loop + continuous retraining pipeline	Q1 2027
Multi-language and multi-channel expansion	Q2 2027

Success Metrics

Tier	Metric	Baseline	Target
North Star	% of verified spam messages intercepted before user delivery	Not tracked	≥ 95% (Recall on spam class)
Primary	Model test precision on spam class	Industry blocklist: ~70%	≥ 93.5% (Decision Tree benchmark)
Primary	False negative rate (spam reaching users)	Not tracked	≤ 8.5%
Secondary	False positive rate (legitimate messages blocked)	Not tracked	≤ 6.5%
Secondary	User-reported misclassification rate	—	≤ 2% of classified messages
Health	Classification latency per message	—	< 100ms P99
Health	Model accuracy on monthly retraining cycle	96.59% (test)	≥ 95% maintained

2. Problem Definition

2.1 Customer Problem

- **Who:** IT security managers and CISOs at mid-to-large enterprises, telecom operators running SMS filtering infrastructure, and mobile app developers embedding SMS validation in their platforms — all responsible for protecting users and employees from cyber threats delivered via text
- **What:** They cannot reliably distinguish malicious SMS (phishing, smishing, prize fraud, malware links) from legitimate business communications at the speed and volume that modern SMS delivery requires. Manual review is infeasible; keyword blocklists are defeated within days as spammers adapt
- **When:** Every second of every day — SMS is a real-time channel; a phishing link delivered and clicked within 30 seconds of receipt has already done its damage before any manual review could intervene

- **Where:** Telecom network filtering layers (before delivery), enterprise mobile device management (MDM) systems, SMS API middleware, and end-user mobile applications
- **Why:** Spam constitutes 13.5% of SMS traffic (validated in 5,572-message dataset: 747 spam vs. 4,825 ham). Spammers actively evolve their vocabulary, use abbreviations, Unicode lookalikes, and sentence structures that evade static rules — making deterministic filtering a losing arms race
- **Impact:** Phishing via SMS (smishing) cost businesses $54.2B globally in 2023 (FBI IC3 report). A single successful phishing link clicked by an employee can result in credential theft, ransomware deployment, and data breaches worth millions. The cost of a missed spam message (false negative) far exceeds the cost of blocking a legitimate one (false positive).

2.2 Why Rule-Based Systems Fail — The Case for NLP and ML

Keyword blocklists and regex-based spam filters fail structurally for five reasons:

1. **Adversarial vocabulary evolution:** Spammers actively probe and adapt to keyword filters. “FREE” becomes “FR33” or “F-R-E-E”; “WINNER” becomes “W1NNER” or “W!NNER”; “click” becomes “cl1ck” or is embedded in shortened URLs. No static ruleset can enumerate all variants; the model learns semantic patterns that survive surface-level mutations.

2. **Context-dependent classification:** The word “call” appears in 480 documents in this dataset — in spam (“CALL NOW to claim your prize!”) and in ham (“Call me when you get home”). Classification requires understanding the full message context, word co-occurrence patterns, and linguistic intent — not the presence of individual keywords.

3. **Class imbalance handling requires probabilistic modelling:** With 86.5% ham and 13.5% spam in the dataset, a rule that classifies everything as ham would achieve 86.5% accuracy while catching zero spam. The model must learn to identify the minority class (spam) with high recall despite the imbalance — requiring statistical weighting (SMOTE, precision/recall optimization) that rule systems cannot provide.

4. **Sentiment and linguistic signal extraction:** Spam messages have a distinctive sentiment profile — SentiWordNet analysis showed spam messages cluster in a distinct distribution vs. ham — and use specific linguistic patterns (urgency, financial reward, action verbs with imperatives). TF-IDF captures these statistical patterns across the corpus; keyword rules cannot.

5. **Scale and speed:** A telecom operator processes millions of SMS per hour. ML inference at < 100ms per message is achievable via a trained model API; equivalent keyword-rule maintenance at scale requires a dedicated team of analysts continuously updating blocklists, a human cost that compounds indefinitely.

2.3 Unstructured SMS Text — Why NLP Is Required

The core input to this system is **raw unstructured SMS text** — free-form, colloquial, abbreviated, and linguistically diverse. This is fundamentally different from structured tabular data, and standard pattern-matching techniques fail on it:

Input Type	Format	Classification Challenge
Promotional spam	Free-text SMS	Prize language, urgency words, financial offers — vary infinitely in phrasing
Phishing SMS	Free-text with URLs	Malicious links embedded in natural-sounding sentences; URL shorteners obscure destinations
Smishing (SMS phishing)	Free-text impersonating legitimate senders	“Your bank account has been suspended” — identical to a legitimate bank notification in structure
Ham (legitimate)	Conversational free-text	Informal language, abbreviations, dialects that look similar to spam without context
Mixed-language SMS	Multilingual / code-switching	Spam in non-English or mixed scripts defeats English-only keyword filters entirely

Why keyword matching fails on raw SMS text: - Abbreviations and shorthand: “u”, “ur”, “txt”, “2nite” — standard tokenisers and keyword lists miss these - Obfuscated characters: “fr33”, “cl!ck”, “w1n” — pattern matching on canonical words misses mutations - Semantic intent requires whole-message understanding: “You have been selected” is spam; “I have been selected for the role” is ham — indistinguishable by word match - Sentence-level TF-IDF captures the relative rarity of prize/urgency vocabulary across the full corpus — a statistical signal that keyword rules physically cannot represent

NLP pipeline applied: 1. Text preprocessing: Lowercase conversion, punctuation removal, stop word removal, tokenisation, stemming/lemmatisation 2. Feature extraction: TF-IDF vectorisation (most frequently discriminating spam terms: “FREE”, “CALL”, “WIN”, “CLAIM”, “PRIZE”, “URGENT”, “TEXT”) 3. Sentiment scoring: SentiWordNet (spam corpus sentiment: bimodal, different from ham’s near-zero distribution) 4. Classification: Decision Tree on TF-IDF feature matrix (5,572 training examples)

Unstructured signal roadmap (V2.0): Future model versions could incorporate additional unstructured signals: sender metadata text (display names, sender IDs), URL destination content (fetching and classifying landing page text), and multi-language SMS using multilingual LLM embeddings — enabling detection of smishing attacks that switch languages or use non-Latin scripts.

2.4 Differentiation from ChatGPT, Copilots, and General AI

“Can’t we just run SMS messages through ChatGPT and ask if they’re spam?” This approach fails for production SMS filtering for four specific reasons:

1. **Latency incompatibility:** ChatGPT API calls take 1–5 seconds per message. Production SMS filtering must operate at < 100ms to avoid delivery delays at telecom-scale volume (millions of messages/hour). A trained Decision Tree classifier scores a message in < 1ms.

2. **Cost at scale:** ChatGPT API pricing at $0.002–0.015 per 1K tokens means classifying 1 million SMS per day costs $2,000–15,000/day — economically unviable for telecom infrastructure. A deployed ML model costs fractionally per million classifications.

3. **No domain-specific training:** ChatGPT was not trained on labelled spam/ham SMS corpora. The Decision Tree model trained specifically on 5,572 labelled SMS messages learns the precise lexical and statistical patterns of spam in this communication medium — outperforming a general model on domain-specific precision.

4. **No API integration or real-time pipeline:** ChatGPT and Copilot are interactive conversational tools, not classifiers that can be embedded in a message delivery pipeline. A production spam filter must intercept messages before delivery, return a binary classification (spam/ham) and a confidence score, and apply this synchronously within the message routing infrastructure — a pattern that requires a deployed ML endpoint, not a conversational AI.

2.5 Defensible Competitive Moat

The SMS Spam Detection System's long-term value compounds through three mechanisms:

1. **Adversarial learning flywheel:** Each user-reported misclassification (false positive or false negative) is a labelled training example. As the system processes real-world SMS and collects feedback, it accumulates a proprietary, time-stamped corpus of labelled spam that reflects current spamming techniques — data no static commercial filter can acquire. Models retrained on this data continuously improve at detecting novel attack patterns.

2. **Domain-specific vocabulary model:** The TF-IDF model learns the specific vocabulary distribution of the organisation's SMS traffic — industry-specific legitimate terms, regional language patterns, and the organisation's own sender IDs. Generic spam filters that are not trained on specific traffic will generate higher false-positive rates on niche vocabulary.

3. **Integration depth and switching cost:** Once embedded in the telecom or enterprise MDM routing layer, the classifier becomes part of the message delivery pipeline. Any replacement requires re-integration, re-training, and re-validation — creating high switching costs that protect the platform.

2.6 Market Context

- **SMS phishing (smishing) financial losses:** $54.2B globally in 2023 (FBI IC3)
- **Global SMS volume:** 2.3 trillion messages per year (Statista 2024); 13.5% spam rate = ~310 billion spam SMS annually
- **Enterprise cybersecurity market:** $202B in 2023, growing at 12.3% CAGR — SMS filtering is a growing sub-segment

- **Spam detection dataset:** 5,572 labelled messages; 4,825 ham (86.5%) / 747 spam (13.5%) — confirmed class imbalance that models must handle
- **Why now:** Smishing attacks increased 328% between 2020 and 2023 (Proofpoint); mobile-first business communication creates a growing attack surface that traditional email spam filters do not cover

---

3. Solution Overview

3.1 What We're Building

A real-time SMS classification API that intercepts every incoming message, preprocesses the text (tokenise → lowercase → remove stop words → TF-IDF vectorise), scores it using a trained Decision Tree classifier, and returns a binary label (spam/ham) plus a confidence score — all in < 100ms. The API integrates with telecom routing infrastructure, enterprise MDM platforms, and mobile application SMS APIs to block spam before delivery. A user feedback interface captures misclassification reports, and a monthly retraining pipeline incorporates new labelled examples to adapt to evolving spam patterns.

3.2 User Flows

Flow 1 — Real-Time Message Classification

> Incoming SMS received by telecom / enterprise gateway → Message text sent to Classification API (REST call) → Preprocessing pipeline: lowercase → punctuation removal → stop word removal → tokenisation → TF-IDF vectorisation → Decision Tree classifier scores message → returns {label: "spam"|"ham", confidence: 0–100%} → Gateway routing decision: if spam → quarantine/block; if ham → deliver → Classification result + timestamp logged to audit trail → Total latency < 100ms

Flow 2 — User-Reported Misclassification (Feedback Loop)

> User receives a message incorrectly classified (false positive: ham blocked; false negative: spam delivered) → User flags message via feedback button in mobile app or email portal → Feedback record created: {message_text, original_classification, user_label, timestamp, sender_id} → Feedback queued for analyst review → Analyst confirms or rejects user label → Confirmed corrections added to retraining queue → Feedback rate tracked on admin dashboard (target ≤ 2% misclassification rate)

Flow 3 — Monthly Model Retraining

> Monthly trigger: pull all confirmed feedback corrections + last 30 days of classified messages → Retrain Decision Tree on expanded corpus (original 5,572 + new labelled examples) → Evaluate new model vs. current model on holdout test set → If new model precision ≥ 93% AND recall ≥ 90%: promote to production → If accuracy degrades: hold for manual review; alert ML team → Version-tagged model deployed to API; prior version archived with rollback capability

Flow 4 — Admin Monitoring and Alerting

Real-time dashboard tracks: messages classified per hour, spam rate, false positive rate, model confidence distribution → Alert fired if: spam rate spikes > 20% (new campaign detected), false positive rate rises > 8% (model degrading), API latency > 200ms P95 → Weekly report: top new spam phrases detected, misclassification patterns, precision/recall trends

Flow 5 — Human-in-the-Loop Review (Low Confidence)

Messages with model confidence 40–60% flagged as "uncertain" (borderline cases) → Uncertain messages routed to human analyst queue (not auto-blocked) → Analyst reviews and labels within 4 business hours → Analyst decision applied; message delivered or quarantined accordingly → Labelled uncertain messages added to training queue immediately → Target: < 2% of total messages fall in uncertain zone after 6 months

3.3 In Scope

Feature	Priority	Description
Real-time classification API	P0	REST endpoint: input SMS text, output {spam/ham, confidence %} in < 100ms
TF-IDF feature extraction pipeline	P0	Preprocessing + vectorisation run on every incoming message
Decision Tree classifier	P0	Trained model with 96.59% test accuracy, 93.50% test precision
Audit logging	P0	Every classification logged with message hash, label, confidence, timestamp
Admin monitoring dashboard	P0	Real-time spam rate, false positive rate, model accuracy, latency metrics
User misclassification reporting	P1	In-app or email button to flag wrongly classified messages
Monthly automated retraining pipeline	P1	Retrain on accumulated feedback; accuracy gate before promotion
Human-in-the-loop review queue	P1	Uncertain-confidence messages routed to analyst
Alert system	P1	Spike detection for spam campaigns, model degradation, latency
Multi-language SMS support	P2	Extend TF-IDF vocabulary to non-English and code-

Feature	Priority	Description
		switched SMS
Sender metadata enrichment	P2	Incorporate sender ID reputation scores alongside text classification
Ensemble model (Decision Tree + Random Forest)	P2	Combine models for higher-confidence predictions on borderline cases

3.4 Out of Scope

- Email spam classification (different medium with different feature vocabulary — separate product)
- MMS / RCS / WhatsApp spam filtering — Phase 2
- Real-time URL reputation scoring (fetching and classifying landing pages) — Phase 2
- Sender identity verification (e.g., SIM-swapping detection) — Phase 3
- Voice/robocall detection — separate product line

3.5 MVP Definition

- **Core Features:** Classification API, preprocessing pipeline, Decision Tree model, audit logging, admin monitoring
- **Success Criteria:** < 100ms P99 latency; ≥ 93% precision on live data; ≥ 90% recall; ≤ 2% API error rate
- **MVP Target:** Q4 2026
- **Learning Goals:** Validate precision/recall on real-world SMS traffic (vs. academic dataset); measure false positive rate impact on legitimate message delivery; validate < 100ms latency under production load

---

4. User Stories & Requirements

4.1 User Stories

Story 1 — Phishing Prevention As an **IT security manager at an enterprise**, I want to **automatically block SMS messages containing phishing content before they reach employees' phones**, So that **employees cannot accidentally click malicious links, preventing credential theft and ransomware incidents**.

Acceptance Criteria: - [ ] Every incoming SMS is classified before delivery (no messages bypass the filter) - [ ] Messages classified as spam are quarantined, not deleted — accessible via admin portal for review - [ ] Classification decision returned in < 100ms so delivery is not perceptibly delayed for ham messages - [ ] Audit log records every decision with message hash (not plaintext) for compliance purposes

---

Story 2 — False Positive Recovery As a **business operations manager who received a blocked legitimate SMS**, I want to **report the misclassification and have the message released within 1 business hour**, So that **urgent legitimate business communications are not permanently blocked and I can trust the system**.

Acceptance Criteria: - [ ] Misclassification report submittable in < 3 clicks via mobile app or email - [ ] Quarantined ham messages released to inbox within 1 business hour of confirmed report - [ ] User receives confirmation when their reported message is reviewed - [ ] Reported cases logged and added to retraining queue within 24 hours

---

Story 3 — Telecom-Scale Filtering As a **telecom engineer managing SMS gateway routing**, I want to **integrate the spam detection API into our message routing pipeline with sub-100ms response time**, So that **spam filtering operates transparently within our delivery SLA without adding perceptible latency for end users**.

Acceptance Criteria: - [ ] API available as REST endpoint with JSON request/response - [ ] P99 latency ≤ 100ms under load of 10,000 messages/minute - [ ] API returns graceful fallback (deliver with “unclassified” flag) if classifier service is unavailable - [ ] Horizontal scaling support: stateless API with no per-request session storage

---

Story 4 — Security Team Visibility As a **CISO wanting to understand the SMS threat landscape facing my organisation**, I want to **see a real-time dashboard of spam volume, attack patterns, and model performance**, So that **I can identify emerging phishing campaigns targeting my company before they cause widespread damage**.

Acceptance Criteria: - [ ] Dashboard shows hourly spam rate, top spam phrases, confidence distribution, precision/recall trend - [ ] Alert fires within 15 minutes if spam rate spikes > 20% above 7-day rolling average (potential campaign) - [ ] Weekly email report automatically generated with key security metrics - [ ] Export to CSV for integration with SIEM systems

---

4.2 Functional Requirements

ID	Requirement	Priority	Notes
FR1	Classification API must return {label, confidence_pct} in < 100ms P99	P0	Core SLA; blocking real-time delivery if breached
FR2	Every classification must be logged with: message_hash (SHA-256), label, confidence, timestamp, sender_id	P0	Compliance and audit trail; no raw message text stored

ID	Requirement	Priority	Notes
FR3	Spam messages must be quarantined (not deleted) and accessible to admin for 30 days	P0	False positive recovery pathway
FR4	False positive report must release quarantined message within 1 business hour	P1	User trust maintenance
FR5	Model must be retrained monthly on accumulated feedback; accuracy gate ≥ 93% precision before promotion	P1	Continuous improvement
FR6	Low-confidence messages (40–60% confidence) must be routed to human analyst queue, not auto-blocked	P1	Human-in-the-loop for uncertain cases
FR7	Dashboard must alert within 15 minutes of spam rate spike > 20% above 7-day rolling average	P1	Campaign detection
FR8	API must degrade gracefully: if classifier unavailable, deliver message with “unclassified” flag (do not block)	P0	Reliability — availability > spam blocking when forced to choose
FR9	Model must handle class imbalance: precision and recall both reported separately for spam and ham classes	P0	Accuracy alone is insufficient given 86.5%/13.5% class split
FR10	All message text must be processed in-memory only; no raw SMS content written	P0	Privacy requirement

ID	Requirement	Priority	Notes
	to permanent storage		

4.3 AI System Capabilities and Autonomy Boundaries

Operates autonomously (no human approval): - Classifying every incoming message as spam or ham - Quarantining spam-classified messages before delivery - Delivering ham-classified messages without delay - Generating daily/weekly monitoring reports - Monthly retraining pipeline execution (up to accuracy gate check)

Requires human approval: - Promoting a retrained model to production (ML engineer reviews accuracy report and approves) - Adjusting the spam classification threshold (security team decision — balances false positives vs. false negatives) - Releasing any quarantined message (admin or user-initiated report required) - Changing the low-confidence boundary (currently 40–60%) that routes to human review - Adding a new sender ID or domain to a permanent allowlist

Error handling and fallback behaviour: - Classifier API timeout (> 100ms): deliver message with “classification timeout” flag; log for batch classification; do not block - Confidence < 40% (extremely uncertain): treat as spam for delivery purposes; route to highest-priority human review; analyst reviews within 2 hours - Model returns error/exception: deliver message with “unclassified” flag; increment error counter; page on-call if error rate > 0.1% - Retraining pipeline produces model with < 90% recall: reject new model; keep current model; alert ML team with detailed performance comparison

4.4 Non-Functional Requirements

- **Performance:** Classification API < 100ms P99; dashboard queries < 2 seconds; retraining pipeline completes within 4 hours on monthly schedule
- **Scalability:** Handle 10,000 classifications/minute at MVP; 1,000,000/minute at telecom scale (horizontal scaling via stateless API replicas)
- **Privacy:** No raw SMS message text persisted to any database; only SHA-256 message hash stored in audit logs; GDPR and CCPA compliant
- **Reliability:** 99.9% uptime for classification API; graceful degradation to deliver-with-flag rather than block on failure
- **Security:** API requires authentication (API key or mTLS); no plaintext messages in logs; model weights protected from extraction attacks
- **Compliance:** GDPR Article 22 compliance for automated blocking decisions; user must have access to contest a spam classification; message hashes in audit log for legal hold requests
- **Explainability:** High-confidence spam decisions must be accompanied by top 3 contributing TF-IDF terms (e.g., “Classified as spam because: FREE, WINNER, CLAIM”) — supports user appeal and analyst review

5. Model Selection and Performance

5.1 Full Model Comparison

Four models evaluated on 5,572 labelled SMS messages (70/30 train/test split):

Model	Train Accuracy	Test Accuracy	Train Recall	Test Recall	Train Precision	Test Precision
Decision Tree (before pruning)	97.22%	**96.59%**	92.03%	**91.50%**	95.71%	**93.50%**
Decision Tree — Pruned	96.75%	94.79%	91.76%	89.90%	93.96%	88.18%
Random Forest (before pruning)	95.83%	95.42%	84.45%	82.89%	97.70%	**97.49%**
Random Forest — Pruned	97.49%	94.70%	91.70%	85.31%	97.32%	90.78%

5.2 Recommended Production Model: Decision Tree (Before Pruning)

Rationale: For a cybersecurity use case, the critical metric balance is: - **Recall (spam class):** Must be high — missing spam (false negative) means a phishing message reaches the user - **Precision (spam class):** Must be high — over-blocking (false positive) erodes trust in the system and harms business communication

The Decision Tree before pruning achieves the best balance: - **Test Recall: 91.50%** — second only to Decision Tree pruned (89.90%) but best overall considering both metrics - **Test Precision: 93.50%** — highest among all models except Random Forest which sacrifices recall (82.89%) for precision (97.49%)

The Random Forest achieves higher precision (97.49%) but significantly lower recall (82.89%) — meaning it misses 17.1% of spam messages. For a security use case, allowing nearly 1-in-6 spam messages through is unacceptable.

The key trade-off justified: Decision Tree precision 93.50% means 6.5% of ham messages may be flagged as spam — a false positive rate acceptable because: (a) quarantine (not delete) means no messages are lost, and (b) the user feedback pathway releases false positives within 1 business hour.

5.3 Key Classification Signals (TF-IDF Top Discriminating Features)

Top spam-indicative terms from TF-IDF analysis and word cloud: - **High-weight spam terms:** FREE, WIN/WINNER, CLAIM, PRIZE, CASH, URGENT, TEXT TO [number], CALL NOW,

BONUS - **High-frequency total terms (both classes):** CALL (480 docs), GET (399 docs), COME (264 docs) — context-dependent, cannot be ruled by keywords alone - **Sentiment signal:** Spam distribution shows bimodal sentiment pattern (urgent positive/negative) vs. ham's near-zero neutral distribution — SentiWordNet overall corpus score +0.173

---

6. Go-to-Market Strategy

Deployment Plan

- **Shadow mode pilot (Q3 2026):** Classify messages but deliver all (no blocking); measure precision/recall on live traffic vs. academic dataset; tune threshold before blocking is enabled
- **MVP live — enterprise (Q4 2026):** Enable quarantine for high-confidence spam (≥ 85%); feedback loop activated; human review for borderline messages
- **Telecom operator integration (Q1 2027):** API packaged for telecom gateway integration; scale testing to 1M messages/hour; SLA defined
- **Multi-language expansion (Q2 2027):** Extend training data and feature extraction to cover non-English SMS; address smishing in regional languages

Target Customer Segments

Segment	Use Case	Integration Point
Enterprise IT/Security	Protect employees from smishing	MDM platform, corporate SMS gateway
Telecom operators	Network-level spam filtering	SMS routing infrastructure
Mobile app developers	In-app SMS validation	SDK or REST API call at message receipt
Cyber Solutions companies	White-label spam detection service	API resale to end clients

7. Risks & Mitigations

Risk	Probability	Impact	Mitigation
Adversarial spam evolution — spammers adapt to model features	High	High	Monthly retraining pipeline; user feedback loop; TF-IDF features adapt to new vocabulary in retraining corpus
False positives block critical business SMS (finance, healthcare alerts)	Medium	High	Human-in-the-loop for < 40% confidence; 1-business-hour release SLA; sender

Risk	Probability	Impact	Mitigation
			allowlisting for known trusted senders
Class imbalance causes model to miss emerging spam patterns (new attack vocabulary)	Medium	High	SMOTE or weighted class sampling in retraining; precision and recall tracked separately for spam class
GDPR / privacy violation if raw message content is logged	Low	High	Hash-only storage (SHA-256 message fingerprint, not plaintext); data minimisation policy enforced at API level
Model drift — live spam distribution diverges from training data	High	Medium	Accuracy monitoring on 30-day rolling actuals; auto-alert if precision drops below 88%; trigger early retraining
Low-confidence zone grows over time (model becomes uncertain on new spam formats)	Medium	Medium	Track % of messages in uncertain zone (40–60% confidence); if > 5% of traffic, trigger immediate retraining

8. Metrics Framework

North Star Metric

% of verified spam messages intercepted before user delivery (Spam Recall) Measured via: Monthly audit of confirmed spam messages (from user reports + analyst review) vs. classification outcomes. Target ≥ 95%. This captures the core security value proposition — every spam message that reaches a user is a potential security incident.

Measurement counterfactual: To measure true recall, 5% of messages classified as ham should be routed through a separate analyst sampling pipeline monthly — comparing analyst labels to model labels on a random sample enables unbiased recall estimation without requiring real phishing incidents to manifest.

Full Metric Hierarchy

Tier	Metric	Unit	Target
North Star	% of confirmed spam messages intercepted (spam recall)	%	≥ 95%
Primary	Test precision on spam class (30-day rolling)	%	≥ 93%
Primary	False negative rate (spam delivered to users)	% of classified	≤ 8.5%
Secondary	False positive rate (ham quarantined)	% of classified	≤ 6.5%
Secondary	User-reported misclassification rate	% of classified	≤ 2%
Secondary	Low-confidence message rate (human queue)	% of total	≤ 5%
Operational	False positive release SLA compliance	% within 1 business hour	≥ 99%
Health	API classification latency P99	Milliseconds	≤ 100ms
Health	Monthly retraining success rate	% of runs passing accuracy gate	≥ 95%

Precision threshold justification: ≤ 6.5% false positive rate is acceptable because spam is quarantined (not deleted), feedback pathway restores ham within 1 hour, and the security benefit (93.5% of phishing intercepted) outweighs the inconvenience cost of 6.5% legitimate messages temporarily held. For organisations where this rate is unacceptable (e.g., critical SMS alert systems), the classification threshold can be raised to ≥ 90% confidence for quarantine.

---

9. Timeline & Milestones

Milestone	Date	Deliverables	Success Criteria
Academic model development	May 2024 ✅	Decision Tree (96.59% accuracy), 4-model comparison, TF-IDF pipeline	Course submission validated
Shadow mode pilot	Q3 2026	Live API deployed; no blocking; recall/precision on real traffic measured	Precision ≥ 90% on live SMS traffic

Milestone	Date	Deliverables	Success Criteria
MVP enterprise live	Q4 2026	Quarantine enabled for ≥ 85% confidence; feedback loop active; dashboard live	≤ 2% false positive rate; < 100ms latency
Telecom integration	Q1 2027	Packaged API for gateway integration; scale tested to 1M msg/hr	99.9% uptime at scale; SLA signed
Continuous retraining operational	Q1 2027	Monthly retrain pipeline live; accuracy gate enforced	First retrain cycle completed; model accuracy ≥ 95% maintained
Multi-language expansion	Q2 2027	Non-English TF-IDF extension; multilingual training corpus	≥ 3 additional languages supported

10. Team & Resources

Role	Allocation
Product / Project Owner	Sankar Kumar Palaniappan
NLP / ML Engineer (model + pipeline)	1 × 100%
Backend Engineer (API + routing integration)	1 × 100%
Security Engineer (audit, compliance, GDPR)	0.5 FTE
Analyst (human-in-the-loop review queue)	0.25 FTE (2 hrs/day)

Infrastructure Budget (Monthly):

Category	Monthly (MVP)	Monthly (Telecom Scale)
API hosting (stateless + auto-scaling)	~$200	~$2,500
Model serving infrastructure	~$150	~$1,500
Retraining compute (monthly batch)	~$50	~$200
Monitoring + alerting	~$100	~$400

Category	Monthly (MVP)	Monthly (Telecom Scale)
Total	**~$500/mo**	**~$4,600/mo**

Revenue justification: Preventing one ransomware incident (avg. $4.35M damage, IBM 2023) pays for ~870 years of MVP infrastructure at $500/mo.

---

11. Open Questions

1. **Threshold tuning for different deployment contexts:** Should enterprises serving critical SMS (healthcare alerts, banking OTPs) have a higher confidence threshold (≥ 90%) before quarantine to reduce false positives in high-stakes communication flows?
2. **Privacy-preserving classification:** Can homomorphic encryption or on-device ML enable spam classification without the message text ever leaving the user's device — addressing privacy concerns in jurisdictions with strict data residency laws?
3. **Sender ID as a feature:** Should sender phone number / short code reputation be incorporated as a feature alongside message text? This would require integration with a sender reputation database.
4. **Adversarial robustness testing:** Has the model been tested against adversarial inputs — intentionally obfuscated spam (L33t speak, character substitution, Unicode lookalikes)? A formal red-team evaluation is recommended before production deployment.
5. **Regulatory compliance:** In some jurisdictions, automatically blocking communications without user consent may have legal implications. Legal review required for each target market before deployment.

---

12. Assumptions Made

- The recommended production model is Decision Tree (before pruning) based on best combined precision/recall balance for the security use case
- 5,572-message dataset (86.5% ham / 13.5% spam) is representative of real-world SMS traffic distribution; production accuracy may differ by organisation type and region
- "Quarantine" means messages are held in a retrievable state for 30 days — not permanently deleted — to enable false positive recovery
- The 93.50% test precision and 91.50% test recall figures are from the academic dataset; shadow-mode pilot is required to validate on production traffic before enabling quarantine
- Human-in-the-loop analyst capacity assumes < 5% of messages fall in the uncertain zone (40–60% confidence); if higher, additional analyst capacity is required
- All SMS text is processed in-memory only; no raw message text is written to any persistent storage; compliance with this requirement must be verified by security audit before production launch