Banking Fraud Detection System – Product Requirements Document

1. Executive Summary

Problem Statement

Banking customers face sophisticated SMS/message fraud through:

- **Header Spoofing**: Attackers mimic legitimate bank headers (e.g., AX-HDFC)
- **Regulatory Exploitation**: Fake "mandatory KYC" requests exploiting users' lack of real-time RBI compliance verification
- **Urgency Manipulation**: Creating panic to bypass rational decision-making
- **Volume Challenge**: High-throughput processing required for mass message verification
- **Semantic Evasion**: Pattern matching fails against varied fraud text formulations

Solution Overview

AI-powered real-time fraud detection system combining:

- Multi-model ensemble ML approach for semantic fraud detection
- Real-time RBI regulatory compliance verification
- Header authentication and sender verification
- High-throughput distributed processing architecture
- User-friendly verification interface

2. System Architecture

Technology Stack

- **Backend**: Go (high-throughput processing, API gateway)
- **ML Pipeline**: Python (training, inference)
- **Frontend**: TypeScript/React
- **ML Models**: Transformer-based + ensemble methods
- **ML API**: Hugging Face Inference API
- **Database**: PostgreSQL (main), Redis (caching)
- **Infrastructure**: Docker Compose
- **Message Queue**: RabbitMQ/Kafka for async processing

Core Components

2.1 Backend Services (Go)

- **API Gateway**: REST/GraphQL endpoints
- **Authentication Service**: JWT-based auth
- **Message Processing Service**: High-throughput queue processor
- **Verification Orchestrator**: Coordinates multiple verification checks
- **RBI Compliance Service**: Real-time regulatory check
- **Header Verification Service**: Sender authentication
- **Alert Service**: User notifications

2.2 ML Pipeline (Python)

- **Model Training Pipeline**: Continuous learning
- **Inference Service**: Real-time predictions
- **Feature Engineering**: Extract semantic features
- **Model Registry**: MLflow for versioning
- **Ensemble Coordinator**: Aggregate multiple model outputs

2.3 Frontend (TypeScript/React)

- **Dashboard**: Real-time message analysis
- **Verification Interface**: Simple check mechanism
- **Alert Management**: Review flagged messages
- **Settings & Preferences**: User configuration
- **Educational Module**: Fraud awareness

3. ML Model Architecture

3.1 Multi-Model Ensemble Approach

Primary Models:

Model 1: DistilBERT Fine-tuned for Fraud Classification

- Base: distilbert-base-uncased
- Task: Binary classification (fraud/legitimate)
- Advantage: Fast inference, good semantic understanding
- Training: Financial fraud dataset + synthetic examples

Model 2: RoBERTa for Semantic Analysis

- Base: roberta-base
- Task: Multi-class fraud type detection
- Categories: KYC fraud, link phishing, urgency scam, impersonation
- Advantage: Better contextual understanding

Model 3: Custom LSTM + Attention

- Architecture: BiLSTM + Multi-head attention
- Task: Sequence pattern detection
- Advantage: Captures temporal patterns in message structure

Model 4: XGBoost for Metadata Features

- Input: Non-textual features (sender header, time, URL count, etc.)
- Task: Binary classification
- Advantage: Fast, interpretable, handles structured data well

Ensemble Strategy:

- **Weighted Voting**: Combine predictions with learned weights
- **Stacking**: Meta-learner (Logistic Regression) on model outputs
- **Confidence Thresholding**: Flag uncertain cases for human review

3.2 Feature Engineering

Text Features:

- Token embeddings (from transformers)
- Named entity recognition (bank names, regulatory terms)
- Urgency indicators (keyword extraction)
- URL/link patterns
- Phone number patterns

Metadata Features:

- Sender header authenticity score
- Time of message
- Message length
- Special character ratio
- Link count
- Regulatory keyword presence

Regulatory Features:

- RBI circular compliance check
- Known legitimate sender database match
- Regulatory keyword alignment

3.3 Training Data Strategy

Dataset Sources:

1. Public fraud message datasets
2. Synthetic data generation (GPT-based augmentation)
3. User-reported fraud cases
4. Legitimate bank message corpus
5. RBI circular text corpus

Data Annotation:

- Binary labels: fraud/legitimate

- Multi-class labels: fraud types
- Confidence scores
- Explanation tags

Data Split:

- Training: 70%
- Validation: 15%
- Test: 15%
- Time-based split for temporal validation

4. Key Features

4.1 Real-Time Verification

- Submit message text + sender header
- Multi-model inference (< 500ms response time)
- Fraud probability score + explanation
- Actionable recommendations

4.2 RBI Compliance Check

- Real-time database of RBI circulars
- Keyword matching against legitimate requests
- Timeline verification (is this requirement current?)
- Source verification

4.3 Header Authentication

- Database of legitimate bank sender IDs
- Spoofing pattern detection
- Historical sender reputation score
- Telecom operator verification (future)

4.4 Educational Dashboard

- Common fraud patterns
- Recent fraud trends
- How to verify legitimacy
- Official bank contact information

4.5 Reporting Mechanism

- User reports suspicious messages
- Contributes to training data
- Community flagging system
- Integration with regulatory reporting

5. Performance Requirements

5.1 Throughput

- Handle 10,000+ verifications per second
- Queue-based processing for peak loads
- Auto-scaling based on demand

5.2 Latency

- API response: < 500ms (p95)
- ML inference: < 300ms
- Database queries: < 50ms
- End-to-end verification: < 1 second

5.3 Accuracy

- Precision: > 95% (minimize false positives)
- Recall: > 90% (catch most fraud)
- F1 Score: > 92%
- False positive rate: < 5%

5.4 Availability

- 99.9% uptime
- Graceful degradation during partial failures
- Cached responses for common queries

6. Security & Privacy

6.1 Data Protection

- End-to-end encryption for message text
- PII detection and masking
- Data retention policy (30 days for non-fraud)
- GDPR/data privacy compliance

6.2 API Security

- Rate limiting per user/IP
- JWT authentication
- API key management
- DDoS protection

7. Deployment Architecture

7.1 Docker Compose Services

```
- api-gateway (Go)
- auth-service (Go)
- verification-service (Go)
- ml-inference (Python)
- ml-training (Python)
- frontend (TypeScript/React)
- postgres (main DB)
- redis (caching)
- rabbitmq (message queue)
- nginx (reverse proxy)
```

7.2 Scaling Strategy

- Horizontal scaling for stateless services
- Database replication for read-heavy loads
- Redis cluster for distributed caching
- Load balancer (Nginx)

8. Monitoring & Observability

8.1 Metrics

- Request volume and latency
- Model prediction distribution
- Error rates and types
- Resource utilization
- Queue depth

8.2 Logging

- Structured logging (JSON)
- Centralized log aggregation
- Audit trails for predictions
- Error tracking (Sentry)

8.3 Model Monitoring

- Prediction drift detection
- Feature distribution monitoring
- Model performance metrics
- A/B testing framework

9. Development Phases

Phase 1: MVP (4–6 weeks)

- Single model (DistilBERT)
- Basic Go API
- Simple React frontend
- PostgreSQL setup
- Docker Compose configuration

Phase 2: Enhancement (4–6 weeks)

- Multi-model ensemble
- RBI compliance database
- Header verification
- Advanced UI features
- Performance optimization

Phase 3: Production (4–6 weeks)

- High-throughput processing
- Monitoring and alerting
- Security hardening
- Load testing
- Documentation

10. Success Metrics

10.1 User Metrics

- Daily active users
- Messages verified per day
- User retention rate
- Report submission rate

10.2 Technical Metrics

- Model accuracy on test set
- API latency (p50, p95, p99)
- System uptime
- Error rates

10.3 Business Metrics

- Fraud cases prevented
- User trust score
- Cost per verification
- Revenue (if applicable)

11. Future Enhancements

1. **Mobile SDKs**: Native iOS/Android integration
2. **Browser Extension**: Real-time webpage verification
3. **Telecom Integration**: Sender verification at network level
4. **Multi-language Support**: Regional language fraud detection
5. **Voice Call Fraud**: Extend to voice phishing detection
6. **Federated Learning**: Privacy-preserving model updates
7. **Regulatory API**: Direct integration with RBI systems
8. **Community Network**: Cross-bank fraud intelligence sharing