TextIntel: (NLP based Threat Classifier) By Team -> TechHunters Demo Video Link



Defense analysts receive high volumes of mixed-language communications (Hindi/English/Hinglish).



Manual triage is slow; critical leads can be missed.



Need automated NLP classification into Benign / Suspicious / Critical with clear visual cues.



Must support secure access, auditability, and fast search across historical intel.





- End-to-end NLP pipeline: ingest -> clean -> detect language
- -> multilingual embeddings -> NER -> threat classification



• Supports Hindi, English, and Hinglish with code-mix handling



• Web-based interface for uploading text, viewing colorcoded alerts, and highlighting entities



• Role-Based Access Control (RBAC): Analysts can upload/analyze, Commanders view summaries only



• Feedback loop: analysts can correct model predictions to improve accuracy over time using a retrain logic



Secure storage with encryption and audit logs

Innovation

- Classification for texts through user input and pdf also to classify, and visuals for critical, benign and suspicious messages.
- Hindi/English messages classification support.
- Multi-role dashboards with role access for user, admin and commander
- Analyst-in-the-loop learning (active learning + weak labels) to reduce drift.
- Entity-aware highlighting prioritizes locations, people, orgs, and weapons.
- Privacy-first architecture: encryption at rest + field-level access.

Tech Stack



• Backend: Python (FastAPI), Node.js



• Frontend: React.js, Bootstrap, Axios



• Storage/DB: PostgreSQL

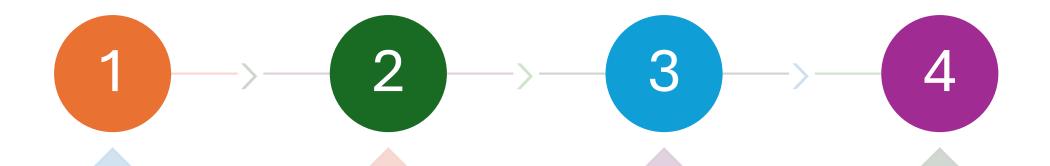


• Auth: JWT



• ML/NLP: SpaCy, Tensorflow, PyMuPDF

Implementation Plan



Phase 1 -> Text classification model using SpaCy and NER to extract entities from the given message. Then, training and saving those models on curated dataset to classify between benign, critical and suspicious.

Phase 2 -> Frontend part using React.js and Bootstrap to build layout for Dashboards, login and register page. Phase 3 -> Role based access logic using JWT tokens to verify based on roles and creating tables to store user data with classified messages and also audit logs to handle user management by admin. Also, the edit and delete messages role by admin.

Phase 4 -> Retrain model logic using subprocess to run retrain script to fetch real-time from db and curated csv dataset to concatenate and train on them. At last, added visuals to see ratio of classified messages for each class with search and sort over the classified messages.

Impact & Future Scope

Fine tuning on multilingual-cased-bert for higher accuracy and more representative embeddings

Analyst productivity via auto-highlighted entities & similar-case context.

Speech to text for intercepted audio; image/doc OCR; geotemporal heatmaps.

Federated/air-gapped training; red-teaming for robustness; continuous drift monitoring.

Vector db for semantic similarity over text embeddings for faster result.