AURA — Autonomous Urban Risk Advisor

Predictive Urban Safety • Dispatch Intelligence

Executive Summary

AURA is an AI-powered urban risk forecasting system that predicts high-risk incidents 30–120 minutes in advance using multimodal data (911/311 logs, CCTV anomalies, traffic, weather, event density, LLM-classified social signals).
The MVP demonstrated a **20–25% reduction in response latency**, **+24% improvement in dispatch efficiency**, and **~82% high-risk recall**, projecting a city-wide operational impact of **~$10M annually**.

---

1. Problem Context

Urban dispatch operations run reactively. Supervisors must manually interpret thousands of signals per hour, resulting in delayed dispatch, misallocated units, and reduced situational awareness.
AURA provides *predictive intelligence*, alert prioritization, and explainability to support faster and more informed decisions.

---

2. Stakeholders & Goals

Stakeholder	Goals
Dispatch Supervisors	Early warnings, priority ranking, dynamic resourcing
911 Operators	Triage assistance, noise reduction
Patrol Commanders	Route optimization, hotspot forecasting
City CIO / Ops	ROI, auditability, fairness compliance

---

3. Key Pain Points

- Average response latency ~24 mins
- Human high-risk recall ~55%
- 18–22% misallocation of units during peak load
- Fragmented data sources and inconsistent real-time visibility

---

4. Multimodal Data Inputs

- 911/311 logs

- CCTV anomaly detection
- GPS + traffic APIs
- Event venues, crowd density
- Weather
- Social text signals (LLM-classified)

---

5. Feature Engineering

- Rolling-window signal densities
- Crowd surge prediction
- Loitering anomaly scoring
- Weather risk modifiers
- Time-of-day sinusoidal encoding
- LLM-based sentiment + topic modeling

---

6. Modeling Approach

AURA uses an ensemble:

1. **XGBoost** for structured features
2. **LLM classifier** for social-text context
3. **Explainability** using SHAP + rationale snippets

MVP Model Metrics:

- Recall (high-risk): **~82%**
- AUC: **0.89**
- Dispatch efficiency: **+24%**

---

7. Architecture

MVP: Python ETL → local feature store → XGBoost → prototype dashboard
Production Design: Kafka → Feast → MLflow → XGBoost + LLM → FastAPI → Dashboard

---

8. Agile MVP Execution — Three Sprints

Sprint 1

- Ingestion simulation
- Feature prototyping
- Data quality checks

Sprint 2

- XGBoost + LLM
- Backtesting
- Explainability (SHAP)

Sprint 3

- Dashboard UX
- Alert priority rules
- Pilot simulation

---

9. KPIs

KPI	Baseline	MVP Projected
Response latency	24 min	18–19 min
High-risk recall	55%	82%
Dispatch efficiency	—	+24%
Yearly savings	—	~$10M

---

10. Simulated Parallel Pilot

No live city systems were accessed.
AURA generated alerts over **historical data streams**, compared against real dispatch logs (control).
This validated predictive uplift, fairness, and operational value safely.

---

11. Business ROI

- Faster responses → fewer secondary incidents
- Better resource allocation → reduced overtime
- Improved route selection → fuel/time savings
- Projected **$10M** per year savings at city scale

---

12. My Contribution

- Full PRD, KPIs, and success metrics
- UX design for dashboard + alert workflows
- Worked with 3 data engineers + 1 ML engineer
- Ownership of pilot design, evaluation, documentation
- Delivered prototype, design system, and portfolio assets

Author

Satya Devi Varaprasad Gundumogula