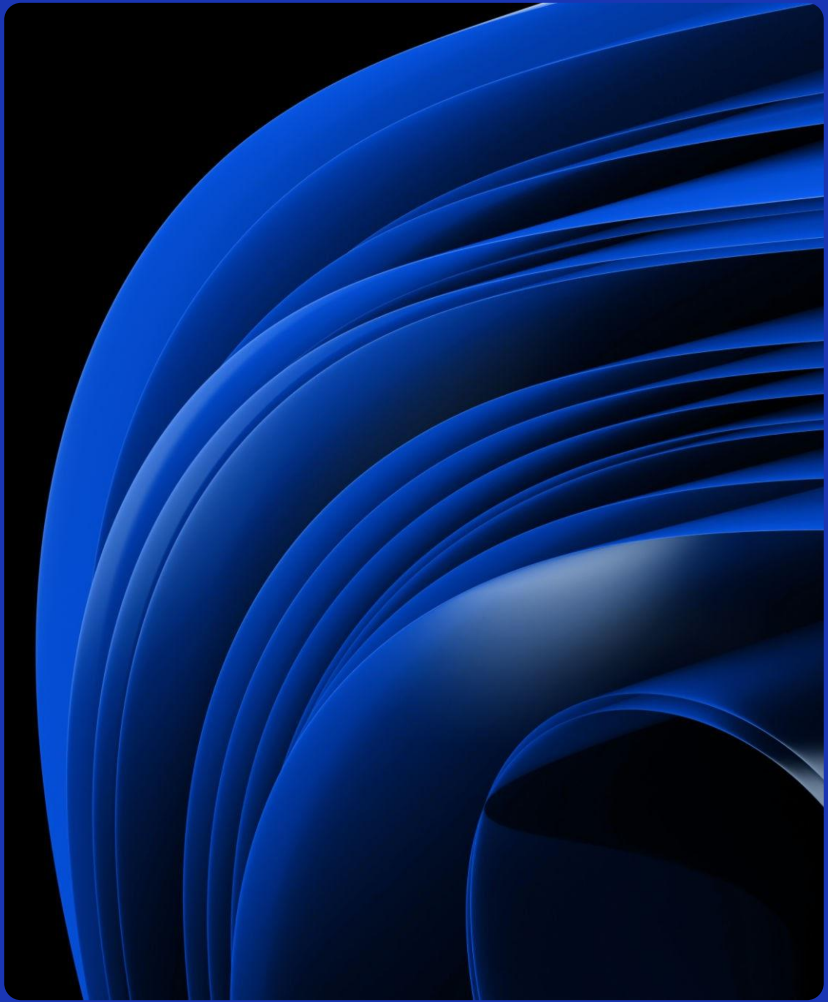
API Anomaly Detection

By: Steven Chua



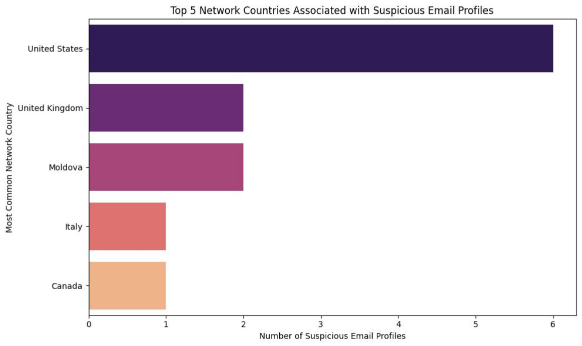
Agenda

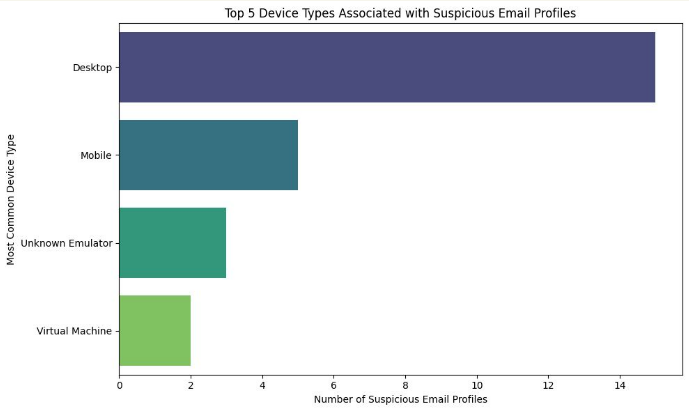
The Challenge: Identifying Suspicious API Users

- **Goal:** Build a model to **classify user profiles** as either **Normal** or **High-Risk/Suspicious** based on their API transaction history.
- **Initial Data Problem (Transaction Level):** The raw dataset had extreme class imbalance: **only 4.3%** of individual transactions were flagged as suspicious.
- **The Solution:** Shift the unit of analysis from a single transaction to the **unique email profile**.
    - **Action:** Aggregate features (e.g., risk_score mean/max, mode of device_type).
    - **Result:** The target distribution improved dramatically: **62.5%** of user profiles were labeled High-Risk (they had at least one suspicious transaction).

Exploratory Data Analysis

1. **Risk Score Metrics are Critical:**
   - `risk_score_max` and `risk_score_mean` are the most powerful numerical predictors. A user's highest single risk score is a strong indicator.
2. **Geographic Concentration:** Suspicious activity is concentrated in specific regions.
   - **Top 5 Network Countries:** US, UK, Moldova, Italy, Canada
3. **Device-Type Signature:** High-risk profiles often share specific device types.
   - **Top 5 Device Types: Unknown Emulator** (the highest indicator), Server, Desktop, Mobile, and Tablet.





Supervised Learning Approach: Finding the Best Classifier

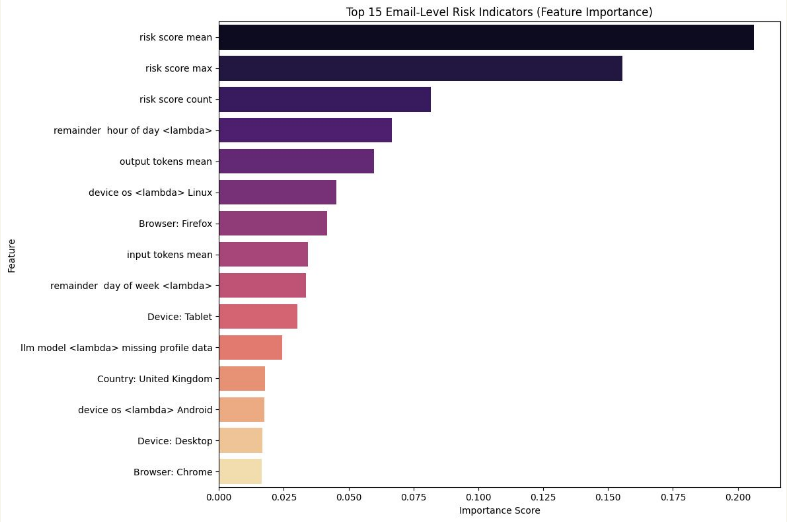
- **Input Data:** Processed, aggregated, and one-hot encoded **Email Profile** dataset.
- **Models Tested:** A comparative analysis of five supervised learning algorithms:
  - **Logistic Regression** (Baseline)
  - **Random Forest**
  - **Gradient Boosting**
  - **SVC**
  - **KNN**
- **Key Metric: ROC AUC Score**
  - Chosen because it measures the model's overall discriminatory power, which is essential for identifying high-risk profiles across all thresholds.

Model	**ROC AUC Score**
Logistic Regression	**1.0000**
Gradient Boosting	**1.0000**
SVC	**1.0000**
Random Forest	0.9857
KNN	0.9286

Feature Importance

Performance Metrics (Random Forest):

- **Overall Accuracy:** 83%
- **Suspicious Class Recall: 100%**
    - This is the critical success metric: the model successfully identified **every single high-risk user** in the test set. (Zero False Negatives for the target class).
- **Confusion Matrix Highlights:** A small number of Normal users were incorrectly flagged as Suspicious (False Positives), which is an acceptable tradeoff for 100% High-Risk detection.



Summary & Next Steps

Takeaways

- We successfully created a robust system for **API anomaly detection** by reframing the problem from transactional logs to **User Profile classification**.
- The aggregated features based on **risk scores, country, and device type** provided highly effective discriminators.

Future Work:

- **Generalizability Test:** Re-evaluate the model on a **larger, blind dataset** to ensure its performance holds up in a real-world scenario.
- **Deployment:** Implement the model in a **live A/B test** environment to measure its direct impact on fraud/anomaly reduction.
- **Deliverable 3:** The full analysis, code, and report are available on my public GitHub repository.

Thank You

Questions?

- **Project GitHub Repository:** https://github.com/stevenchua/api-anomaly/blob/main/api_anomaly.ipynb
- **Video Demo Link:** https://www.loom.com/share/3be1d33c35004ce8b003d1dfe2d2c8a1?sid=7827cacf-ffec-42ac-8c2f-e84bcacb4dea