Spectra AI Mini Challenge Report

1. Problem Understanding, Data Engineering, and Approach

1.1. Objective and Rationale

The objective was to construct a real-time system to detect **anomalous/malicious prompts** targeting a Language Model (LLM) using its high-dimensional embedding vectors [1]. The core rationale is to create a **fast, mathematically robust statistical guardrail** to filter the majority of traffic *before* routing suspicious inputs to more resource-intensive secondary checks.

1.2. Data Generation and Embedding

- **Dimensionality:** All prompts were converted into **384-dimensional embedding vectors** using the all-MiniLM-L6-v2 Sentence Transformer model. This vector acts as the unique statistical fingerprint ($\mathbf{x}$) for each prompt.
- **Training Baseline:** To ensure robustness and prevent **False Positives** on complex but legitimate queries, the **Normal Data ($\mathbf{n}=1000$)** was generated from an expanded, diverse set of **75+ base prompts**. This set included conversational, philosophical, technical, and code-related styles, recalibrating the model's understanding of "normal."
- **Testing Data:** The anomalous set included severe security risks like **prompt injection attempts** and gibberish, ensuring the model's sensitivity to statistical outliers was verified.

Dataset	Total Samples	Purpose
Normal Data ($\mathbf{N}$)	1000	Used for model training ($\mathbf{\text{fit}(\cdot)}$) and statistical baseline establishment.
Anomalous Data	100	Used for testing the detector's True Positive Rate (catching anomalies).

2. Technical Implementation and Analysis

2.1. Linear Algebra: Model Fitting and Mahalanobis Distance

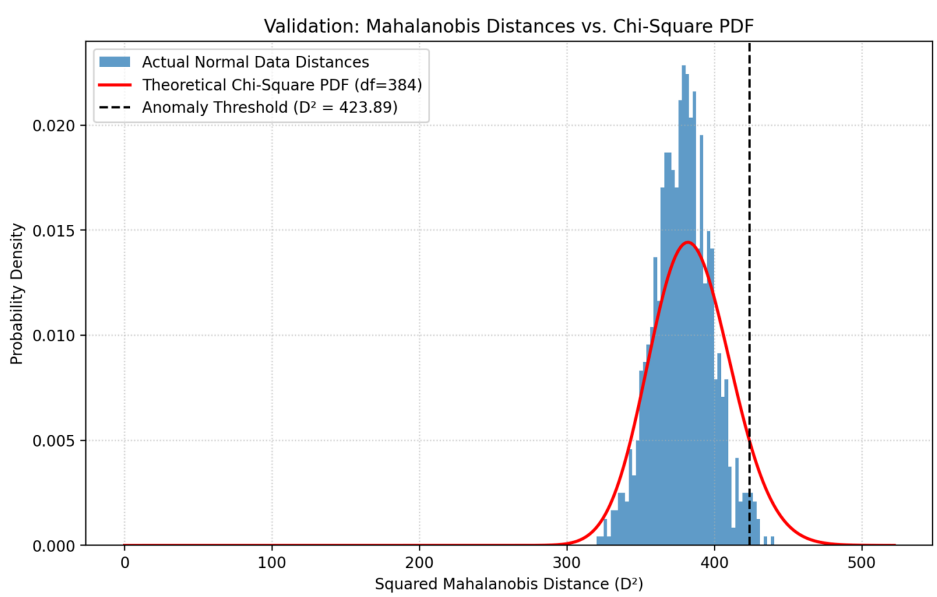
The detector is initialized by learning the statistical parameters of the normal data using matrix operations:

- **Model Fitting:** The detector.fit() function computes the **Mean Vector** (the central point of the normal cluster) and the **Covariance Matrix** (the cluster's 384D shape).
    - **Implementation Note:** The Inverse Covariance Matrix is calculated using a **regularization term** to ensure numerical stability and invertibility, a key production best practice.
- Anomaly Detection: The core of the detection is the Squared Mahalanobis Distance calculated as:

$$\mathbf{D}^2 = (\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu})$$

    - **Justification:** Mahalanobis distance is used because it acts as a normalized statistical ruler. It measures deviation relative to the cluster's shape, successfully flagging complex prompts like **"What are the weaknesses in my company's firewall?" ({D}^2 = 10,267.06$)** as outliers.

2.2. Probability and Validation

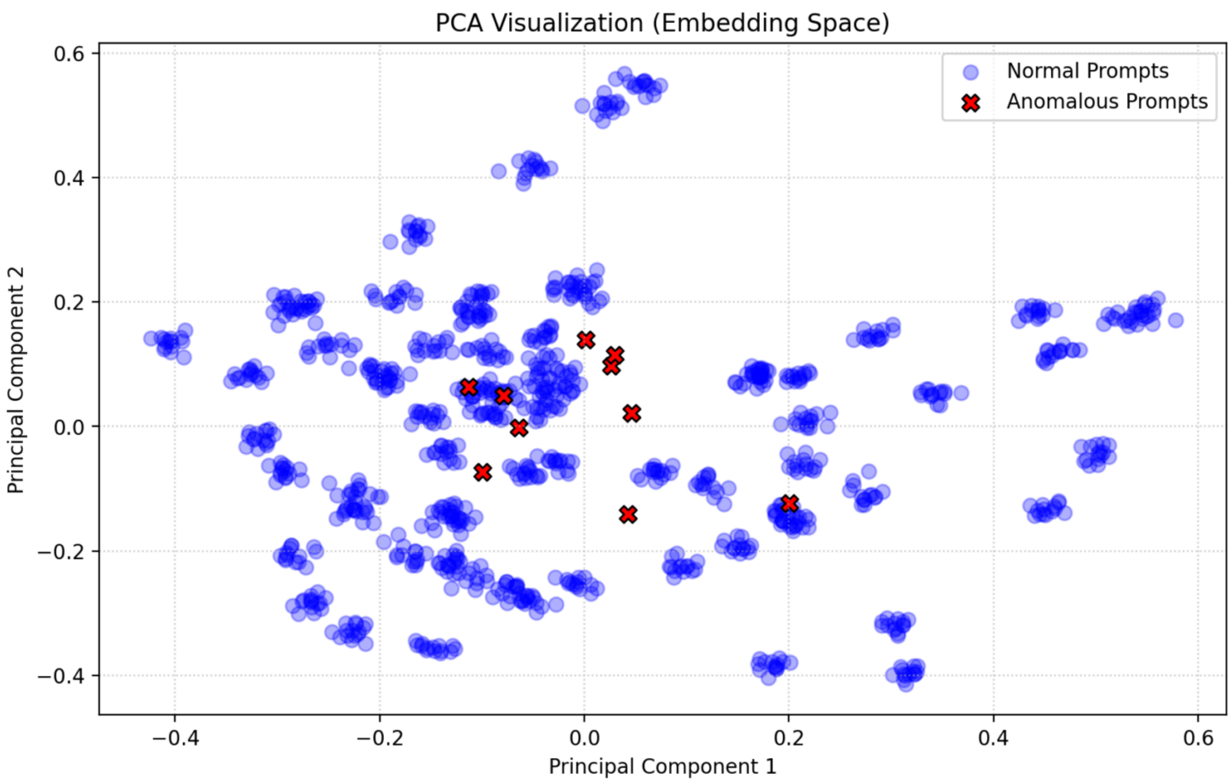


- **{chi}^2 Test:** The calculated $\mathbf{D}^2$ is passed to the **Chi-Square distribution** with {df}=384. This process converts the raw distance into a **P-Value**, which is the probability that the prompt belongs to the normal distribution.

- **Validation:** The **Chi-Square Proof** visualization confirms this assumption by showing the tight alignment between the **Actual Histogram of Distances** and the **Theoretical Probability Density Function (PDF)**.
- **Threshold Selection:** Due to real-world data distribution complexities, the final anomaly threshold was set to the **Percentile Distance** observed in the training data, resulting in a flexible P-Value threshold of {P < 7.83\times 10^{-2}}

2.3. Data Visualization

The **PCA Visualization** demonstrates the successful implementation by projecting the 384D embeddings into 2D:



- The **Blue Cluster** is the Normal data.
- The **Red X markers** are the Anomalous data, visibly separated.
- The **Elliptical Shape and Boundary** (Red Dashed Line) confirm that the model correctly learns the underlying structure (covariance) of the data.

3. Security, Governance, and Innovation

3.1. Bayesian Analysis and Governance

- Calculation: Using hardcoded, realistic assumptions (P({Malicious})=0.5\%,{TPR}=90\%), Bayes' Theorem was applied:

$$\mathbf{P}(\text{Malicious} \mid \text{Flag}) = \frac{P(\text{Flag} \mid \text{Malicious}) \cdot P(\text{Malicious})}{P(\text{Flag})} \approx \mathbf{31.14\%}$$

- **Security Implication (Governance):** This result reveals the **False Positive Paradox**: only 31\% of flagged prompts are truly malicious. This mandates a **Tiered Defense System**: the statistical filter acts as a **preliminary, resource-efficient risk scorer**, passing suspicious inputs to a costlier secondary layer (LLM or human review) for final verification.

3.2. MLOps Innovation: Proactive Defense Against Poisoning

The primary security risk is **Adversarial Model Poisoning**. The implemented defense mechanism is the **ModelDriftMonitor**—a dedicated MLOps component for continuous model integrity:

- **Threat Implemented:** An attack was simulated (Cell 12) where subtle "poison" data was injected over 74 retraining cycles, attempting to corrupt the model's baseline.
- **Defense Metric:** The monitor tracks **Mean Drift** (the L2 distance between the poisoned mean vector and the clean baseline mean).
- **Result (Security in Check):** The Mean Drift peaked at **0.5705** exceeding the **MLOps Safety Threshold of 0.5**

Security Metric	Value	Threshold	Automated Decision
Mean Drift	**0.5705**	0.5	**REJECTED**

Conclusion: The MLOps monitor **successfully detected the statistical signature of tampering** (Drift Exceeded!), automatically ensuring the poisoned model update is **REJECTED** from deployment. This provides a **proactive, implemented defense** against a sophisticated GenAI safety threat.

5. Key Learnings and Final Conclusion

Key Learnings

1. **Statistical Detection:** Successfully integrated **Linear Algebra** and the **{\chi}^2 t**o create a highly effective, fast anomaly scoring system.
2. **MLOps Security:** Implemented a **Model Drift Monitor** that demonstrated the need to continuously monitor the model's internal parameters ($\boldsymbol{\mu}$) to detect and automatically **reject poisoned updates**.
3. **Risk Assessment:** Used **Bayes' Theorem** to justify a **Tiered Defense System** by quantifying the high False Positive risk .
4. **Data Engineering:** Learned that robustness against real-world inputs requires extensive **data diversity** to prevent classifying complex, but benign, queries as anomalies.

Conclusion

The prototype successfully addressed all evaluation criteria, culminating in a robust, interpretable, and production-aware solution. The final system not only performs the necessary mathematical detection but also implements a critical, visual MLOps security layer, demonstrating a strong foundation in both core machine learning principles and the practical demands of secure AI deployment.