AIID Report

**Entity Identification and Relationship Curation in the AI Incident Database**

**Group 2**

**Aiman Muzaffar, MSBA** [**aiman.muzaffar@gwu.edu**](mailto:aiman.muzaffar@gwu.edu)

**Akshay Thirumal Reddy, MSBA** [**akshay.thirumalreddy@gwu.edu**](mailto:akshay.thirumalreddy@gwu.edu)

**Daniel Bazargun, MSBA** [**daniel.bazargun@gwu.edu**](mailto:daniel.bazargun@gwu.edu)

**TABLE OF CONTENTS**

* [**Introduction**](#kix.sz5uqwrmgh4v)
* [**Problem Statement**](#kix.2e2o0agl1sx1)
* [**Methodology**](#kix.2205brqf4cl)
  + [**Data Pipeline**](#kix.rmj6ybklwq06)
  + [**Entity Recognition**](#kix.p0h88hgkxcd)
  + [**Entity Clustering**](#kix.lngaxkxg16of)
* [**Recommendations / Conclusions**](#kix.o3xwdfu7m6iv)
* [**Risk Considerations**](#kix.mtm1y66z2ywa)
* [**Appendix**](#kix.1zjf3daluyfs)

###### **Abbreviation List**

AIID: Artificial Intelligence Incident Database

NLP: Natural Language Processing

ML: Machine Learning

LLM: Large Language Model

NER: Named Entity Recognition

RAG: Retrieval-Augmented Generation

TF-IDF: Term Frequency-Inverse Document Frequency

RoBERTa: Robustly Optimized BERT

K-Means: A clustering algorithm that partitions data into predefined clusters

AWS: Amazon Web Services

ER Diagram: Entity Relationship Diagram

API: Application Programming Interface

#### **1. Introduction**

The **AI Incident Database (AIID)** is an open-source repository documenting real-world incidents where AI systems have caused harm. These incidents involve three key stakeholder groups: developers, deployers, and victims. Understanding the relationships between these entities is crucial to analyzing systemic risks and enabling effective governance.

Currently, the AIID relies on **manual processes** for identifying and clustering entities, which are inefficient and time-consuming. This project focuses on automating these tasks using **data science, natural language processing (NLP)**, and **machine learning (ML)** techniques. The goal is to enhance data quality, derive actionable insights, and establish a scalable framework for future AI incident analyses.

#### **2. Problem Statement**

#### The project addresses the following challenges:

1. **Manual Curation**: Identifying and clustering entities in AI incidents is resource-intensive.
2. **Complex Relationships**: Developers, deployers, and victims are interlinked in evolving, intricate networks.
3. **Limited Insights**: A lack of structured relationships hinders effective analysis and informed decision-making.

**Research Questions**:

* How can automation improve the accuracy and efficiency of entity identification in AI incidents?
* What additional insights can be extracted by curating structured relationships among entities?
* How does structured entity curation enhance analysis, editing, and decision-making?

This project aims to develop tools and methods to address these challenges while ensuring adaptability for future AI-related analyses.

#### **3. Methodology**

The methodology integrates data science, natural language processing (NLP), and machine learning (ML) techniques to streamline entity identification and relationship mapping. Below is a detailed breakdown of the process.

##### **3.1 Data Pipeline**

##### **3.1.1 Data Sources**

The primary data source is the **AI Incident Database (AIID)**, containing detailed reports on over **750 unique AI incidents**. This database is organized into three core collections within MongoDB:

* **Entities Collection**: Includes 2,388 unique entities categorized into developers, deployers, and victims.
* **Reports Collection**: Contains 3,921 indexed reports with metadata and textual descriptions.
* **Incidents Collection**: Documents 793 distinct incidents, linking each to multiple related media reports for context.

##### **3.1.2 Data Preprocessing and Storage**

Preprocessing steps were undertaken to ensure the data is accurate, uniform, and ready for analysis:

1. **Database Management**:
   * MongoDB was chosen for its scalability and compatibility with unstructured data.
   * Python scripts facilitated data querying, organization, and processing.
2. **Standardization**:
   * Text fields were cleaned and standardized for uniformity across collections.
   * Indexed media reports were matched with their respective incidents for streamlined retrieval.
3. **Data Validation**:
   * Validation techniques were applied to maintain accuracy and consistency across the database.

##### **3.2 Entity Recognition**

##### **3.2.1 Natural Language Processing and Machine Learning Techniques**

The project adopted a suite of advanced NLP and ML techniques to identify and cluster entities from reports:

###### **Named Entity Recognition (NER)**

* **Objective**: Extract key entities (developers, deployers, victims) from textual reports.
* **Model Used**: Pre-trained RoBERTa transformer model fine-tuned for entity recognition.
* **Steps**:
  1. **Tokenization**: Converted raw text into smaller units (tokens) for model ingestion.
  2. **Embedding**: Represented tokens in high-dimensional vector spaces.
  3. **Entity Tagging**: Predicted tags for tokens, grouping them into coherent entities.

Named Entity Recognition (NER) was the first approach focusing on extracting, developers, deployers, and victims, from textual incident reports. To achieve this, a pre-trained RoBERTa transformer model, fine-tuned for entity recognition, was used. The process began with tokenization, where raw text was broken down into smaller, manageable units (tokens) to prepare the data for model ingestion. These tokens were then represented in high-dimensional vector spaces through embedding, capturing their semantic meaning. Finally, the model predicted tags for these tokens, grouping them into coherent entities relevant to the incident.

Despite its strengths, the NER process had several challenges. The model struggled to identify abstract entities, such as "pedestrian," or broader collective groups like "residents of a city," as they are more of a concept than a name. Additionally, recognizing victims proved particularly difficult, as the context in reports was frequently ambiguous or insufficiently extracted. Another significant issue was the model's inability to effectively disambiguate entities in scenarios where roles overlapped or were nuanced, leading to misclassification of key entities. These challenges highlight the complexities of applying NER to real-world, unstructured datasets.

**RAG (Retrieval-Augmented Generation) Approach with GPT**

Process:

* Incident Details Query: Constructed a query with title, description, and alleged entities.

GPT Chat Model (RAG):

* Using gpt-4o-mini.
* Asked GPT to extract a single entity for each category (Developer, Deployer, Harmed Parties).

Output Format:

* Developer: <entity>
* Deployer: <entity>
* Victim: <entity>

The Retrieval-Augmented Generation (RAG) approach using GPT API was explored as a method to extract key entities, including developers, deployers, and harmed parties, from AI incident data directly and not from reports. The model was reading the incidents descriptions as the source to do extraction. This process involved constructing detailed queries that incorporated the incident title, description, and alleged entities. Using the GPT-4o-mini model, the system was tasked with extracting a single entity for each category, outputting results in a structured format: Developer, Deployer, and Victim. The implementation was applied row-by-row across the incident database, and the processed results were saved. The results were promising however they were not used as the final approach.

A primary concern was the risk of subjective bias introduced by manually created incident descriptions, which could compromise the accuracy and objectivity of the entities. Additionally, the method was computationally intensive, requiring significant processing power for the GPT API, which made it challenging to use it directly on the reports. Nonetheless, the approach holds substantial potential for future promises, particularly for improving report-based entity extraction once computational and resource constraints are met.

**3.3 Entity Clustering**

**Clustering with Text Embeddings**

To group AI incident entities into meaningful clusters for enhanced analysis and insights.

**3.3.1 Clustering Methodology**

**Text Representations:**

* Text Embeddings: Utilized (all-MiniLM-L6-v2) to generate high-dimensional semantic text representations for the entities.

Clustering with text embeddings was employed to group AI incident entities into meaningful clusters. This approach utilized the all-MiniLM-L6-v2 model to generate high-dimensional semantic representations of text, effectively capturing the underlying context and relationships between entities. These embeddings were applied to entities manually created from the AI Incident Database.

By using these semantic text representations, entities were grouped into clusters that reflected their similarities and relationships within the dataset. This structured clustering approach was integral to uncovering meaningful insights while providing scalability for future analyses.

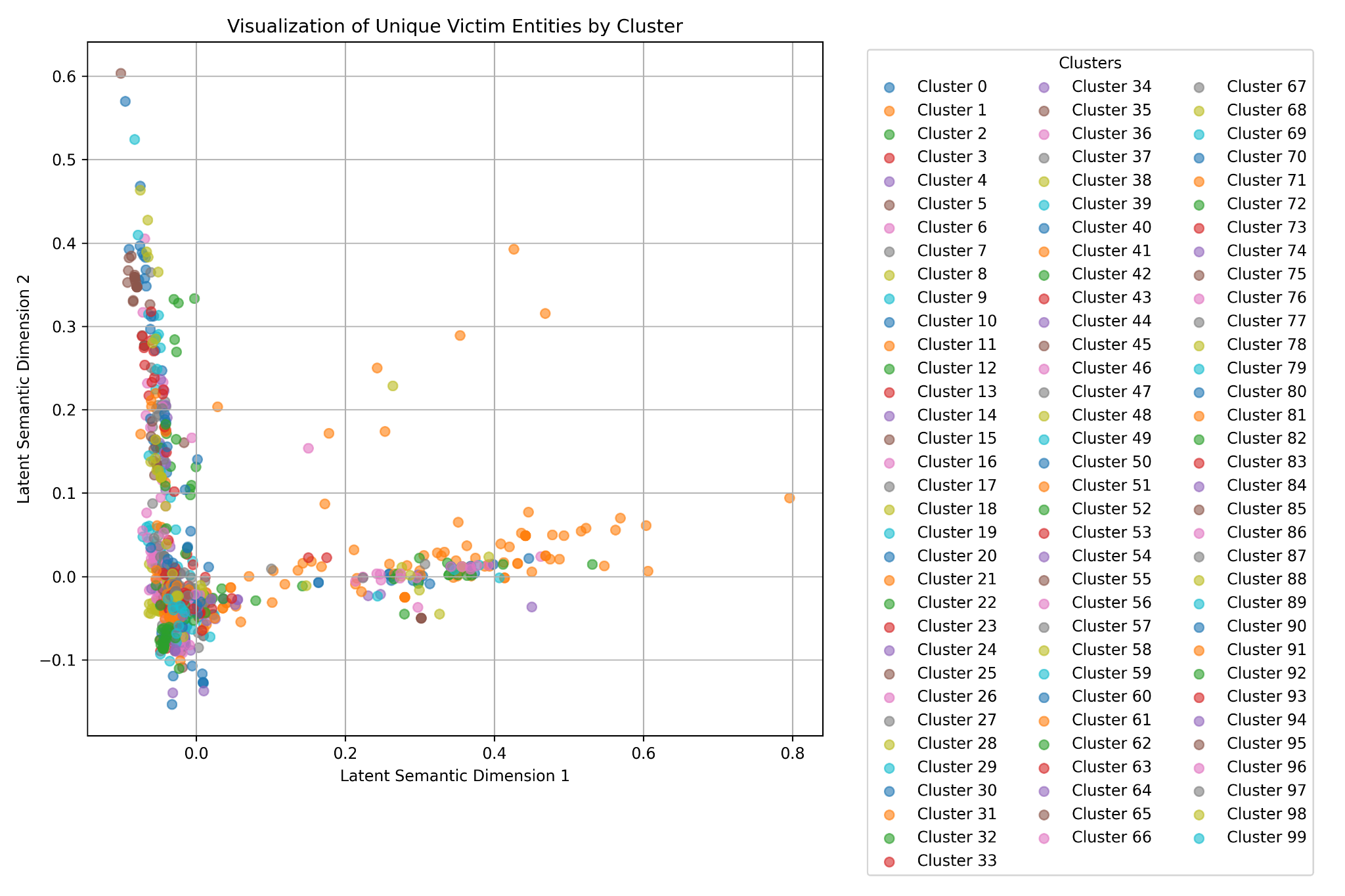
**Clustering Techniques:**

* K-Means Clustering: Partitioned data into predefined clusters based on similarity.

K-Means clustering was applied to these embeddings with a number of clusters set to 100, creating distinct groups of entities based on their semantic similarity. This approach showed the patterns and relationships among developers, deployers, and harmed parties within the dataset. By organizing entities into 100 structured clusters, K-Means clustering simplified the analysis of large datasets and revealed themes,and enhanced the ability to derive actionable insights from the data.

* Successfully created distinct clusters that revealed patterns in incident themes
* Enhanced understanding of relationships and trends within the dataset, providing insights into recurring issues and their implications.

This structured approach not only streamlined analysis but also facilitated a deeper understanding of the "Victim" category in the AI incidents Database.



*Fig 1: Text Embedding Victim K-Mean Scatter Plot*

The scatter plot above shows the clustering of unique victim entities using K-Means, where each point represents an entity embedded in a high-dimensional semantic space and assigned to one of 100 clusters on the right.

**3.3.2 Graphing the Clustered Entities**

| **Node** | **Label** | **Cluster Label** |
| --- | --- | --- |
| Navya | Navya | Autonomous Transit Systems Transforming Public Mobility |
| Tiktok | Tiktok | TikTok Users and Demographics |
| ESPN | ESPN | Media Outlets, News Platforms, and Audiences |

*Table 1: Node Table*

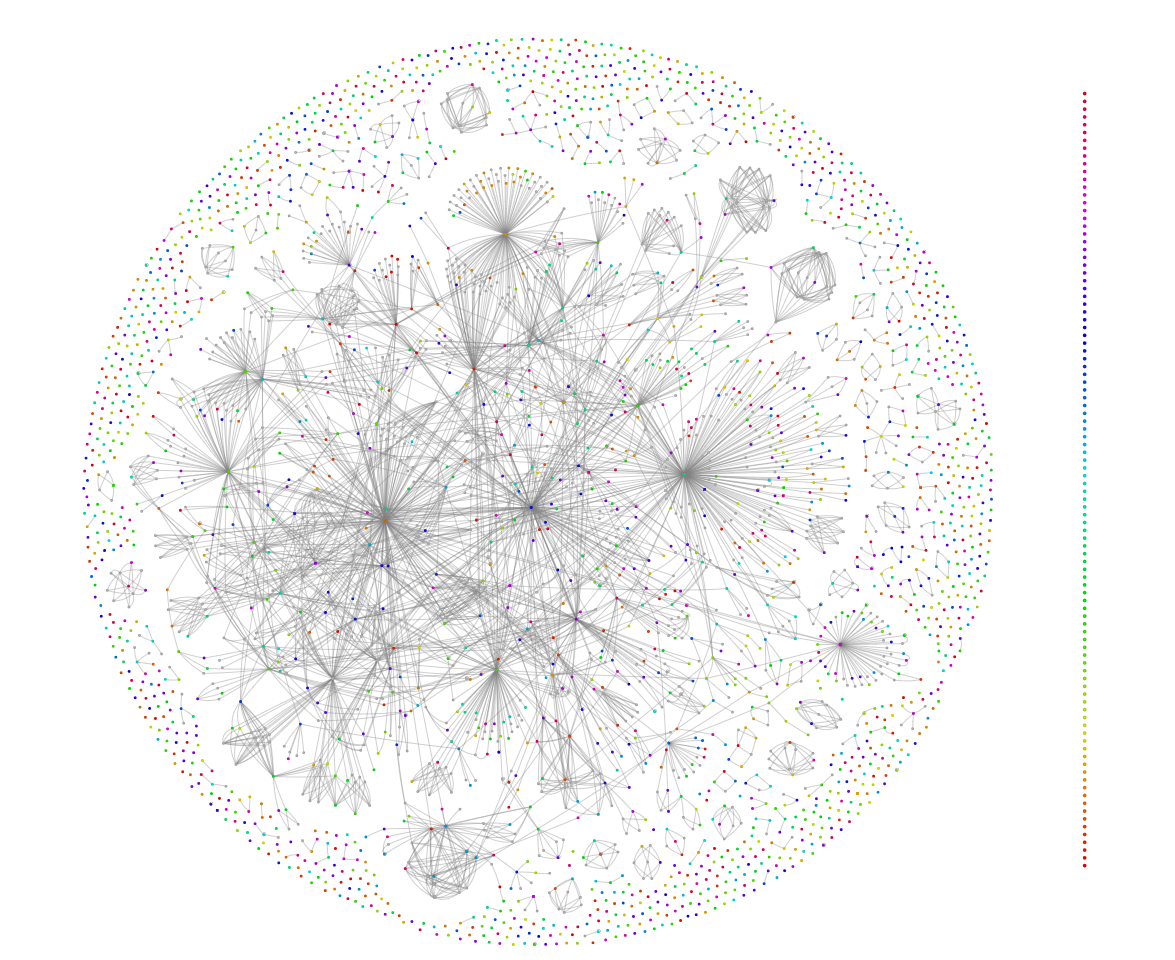
The visualization of clustered entities begins with the structured data derived from the AI Incident Database, where key actors such as developers, deployers, and victims are identified. The clustering process utilizes K-means clustering, to group entities based on their feature similarities. This approach segments the dataset into distinct clusters, as shown in the nodes table, where each node is assigned a cluster number. For example, entities like “Facebook" and "Tesla" are grouped based on shared attributes or their involvement in related incidents.

| **Source** | **Target** | **Relationship** |
| --- | --- | --- |
| tesla | general public | Harms |
| tesla | motorists | Deployed by |
| tesla | joshua brown | Directly harms |
| facebook | Facebook users | Harms |

*Table 2: Edge Table*

The edges, defined in the edges table, establish relationships between these entities, such as "Deployed by”, connecting developers and deployers to affected victims. This combination of clustering and relationship mapping provides a framework for understanding the interconnections within AI incidents and supports stakeholders in uncovering trends or recurring issues in the dataset.

**3.3.3 Entity Relationship Graph**

****

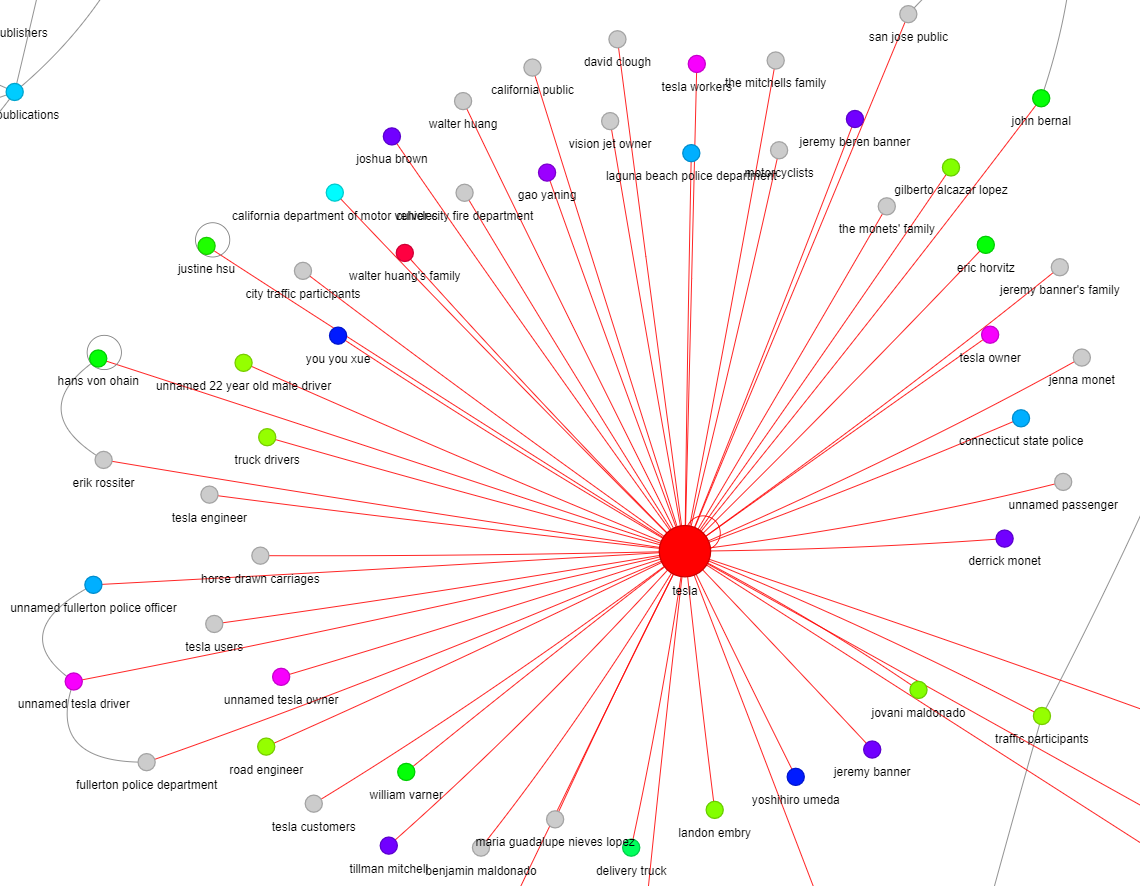
*Fig 2: AIID Network Graph (generated using pyviz and NetworkX)*

The entity relationship graph provides a structured visualization of the relationships among developers, deployers, and victims involved in AI incidents. Each node represents a distinct entity, such as a developer responsible for creating an AI system, a deployer utilizing the system, or a victim affected by its deployment. The edges between nodes capture specific relationships, including:

* **Developer → Deployed by → Deployer:** This relationship links developers to deployers, illustrating how AI systems are transferred and utilized in real-world applications.
* **Developer → Directly harms → Victim:** This edge highlights incidents where harm is caused directly by the AI systems designed by developers.
* **Deployer → Harms → Victim:** This captures cases where deployers' use of AI systems leads to harm, reflecting the broader impact of their operational decisions.

These relationships, coupled with clustering using K-means, allow stakeholders to analyze patterns of responsibility and harm within the AI ecosystem. The graph’s color-coded clusters further group related entities, providing a clear and comprehensive view of the interconnections and their implications. This visualization is a powerful tool for identifying areas where ethical AI governance and accountability are most urgently needed.

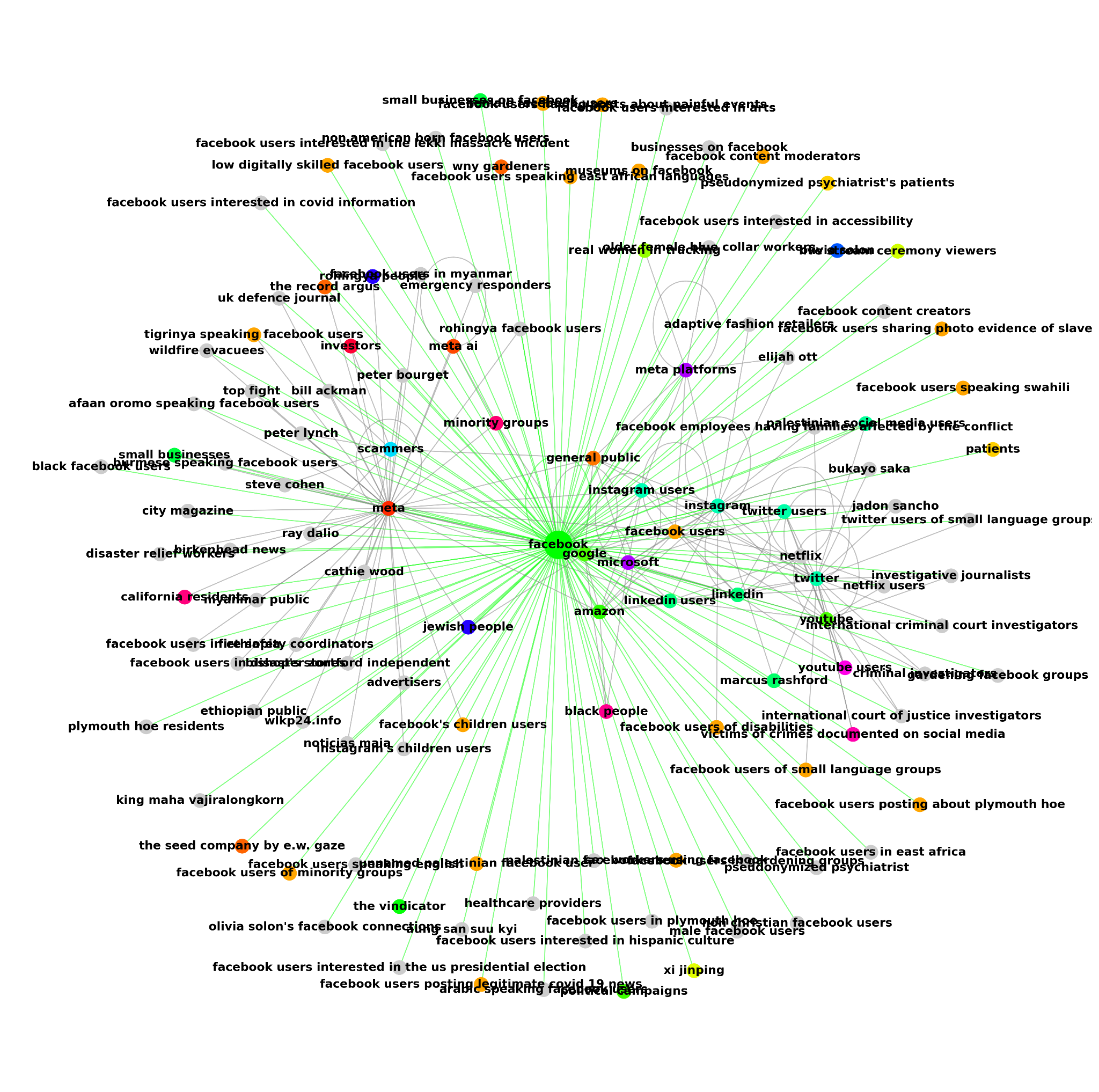
### **Closer Look at Tesla Subgraph**



*Fig 3: Tesla Subgraph*

**Tesla’s** subgraph illustrates the risks associated with autonomous AI systems, particularly self-driving cars. Victims include accident **victims, pedestrians, and engineers**, reflecting the consequences of deploying AI without adequate safeguards. Tesla’s graph underscores the importance of safety protocols and regulatory oversight, highlighting the potential for significant harm in autonomous systems.

### **Closer Look at Facebook Subgraph**



*Fig 4. Facebook Subgraph*

The graph of **Facebook** dives deeper into specific cases. Shows various groups affected by AI systems deployed by Facebook, such as **minority groups, children users, and instagram users.** Highlights how Facebook’s AI systems impact diverse users, including vulnerable populations (e.g., children, minority groups). Raises concerns about moderation, privacy, and algorithmic biases affecting these groups.

#### **4. Recommendations / Conclusion**

##### **4.1 Conclusion:**

**Automated Entity Identification**

The project successfully demonstrated that automating the extraction of developers, deployers, and victims from AI incident reports is feasible, reducing reliance on manual curation and improving data quality.

**Clustering Insights**

The use of text embeddings provided a robust method for grouping entities based on their semantic similarity, allowing for more meaningful and context-aware results. This approach revealed patterns in AI incidents, such as common relationships between deployers and victims. By leveraging embedding techniques, the quality of the clustering process significantly improved compared to simpler methods like TF-IDF, making it possible to uncover more nuanced clusters.

**4.2 Recommendations:**

**Enhance Entity Extraction Models**

* Integrate context-aware language models (e.g., GPT-4) to handle ambiguous terms and improve precision in entity identification.
* Incorporate additional pre-processing techniques to standardize and disambiguate entities.

**Optimize Clustering Algorithms**

* Including features like entity type, frequency, and sentiment enhances clustering by adding context. Temporal and location data further uncover meaningful patterns and trends.
* Pre-processing techniques like synonym replacement, normalization, and context-aware disambiguation ensure clean and consistent data for clustering.

**Improve Computational Power**

* Utilize scalable cloud-based infrastructure, such as AWS or Google Cloud, to optimize processing times, especially for resource-intensive tasks like running large language models (LLMs).

To enhance entity extraction models, combining traditional NLP techniques with advanced large language models (LLMs) can address contextual ambiguities and improve the accuracy of identifying complex entities, especially when trained on domain-specific datasets tailored to the context of AI incidents. Optimizing clustering algorithms can be achieved by enriching input data with additional features, such as contextual data with details, to better capture relationships and trends within the dataset.

Addressing computational limitations is crucial; increasing processing power and implementing distributed computing solutions will significantly reduce processing times for resource-intensive tasks, enabling the system to handle large-scale datasets more efficiently and effectively, especially when it comes to using LLMs directly on the reports.

#### **5****. Risk Considerations**

The project encountered several risks and limitations that impacted various aspects of the work.

**Data Challenges:**

One challenge was the textual descriptions of incidents, which often contained ambiguity, making it difficult to identify abstract entities such as broad groups or conceptual terms. Additionally, errors or biased perspectives present in the manually curated AI Incident Database further complicated the reliability of entity extraction, potentially affecting the overall quality of the results.

**Model Limitations:**

The limitations of the model also presented limits. Incident descriptions that were manually created introduced a risk of subjective bias, which could compromise the extracted entities. Furthermore, the model struggled in complex or nuanced scenarios, often misidentifying or lacking critical entities due to contextual challenges. These issues highlighted the need for improved methods to handle such intricacies effectively.

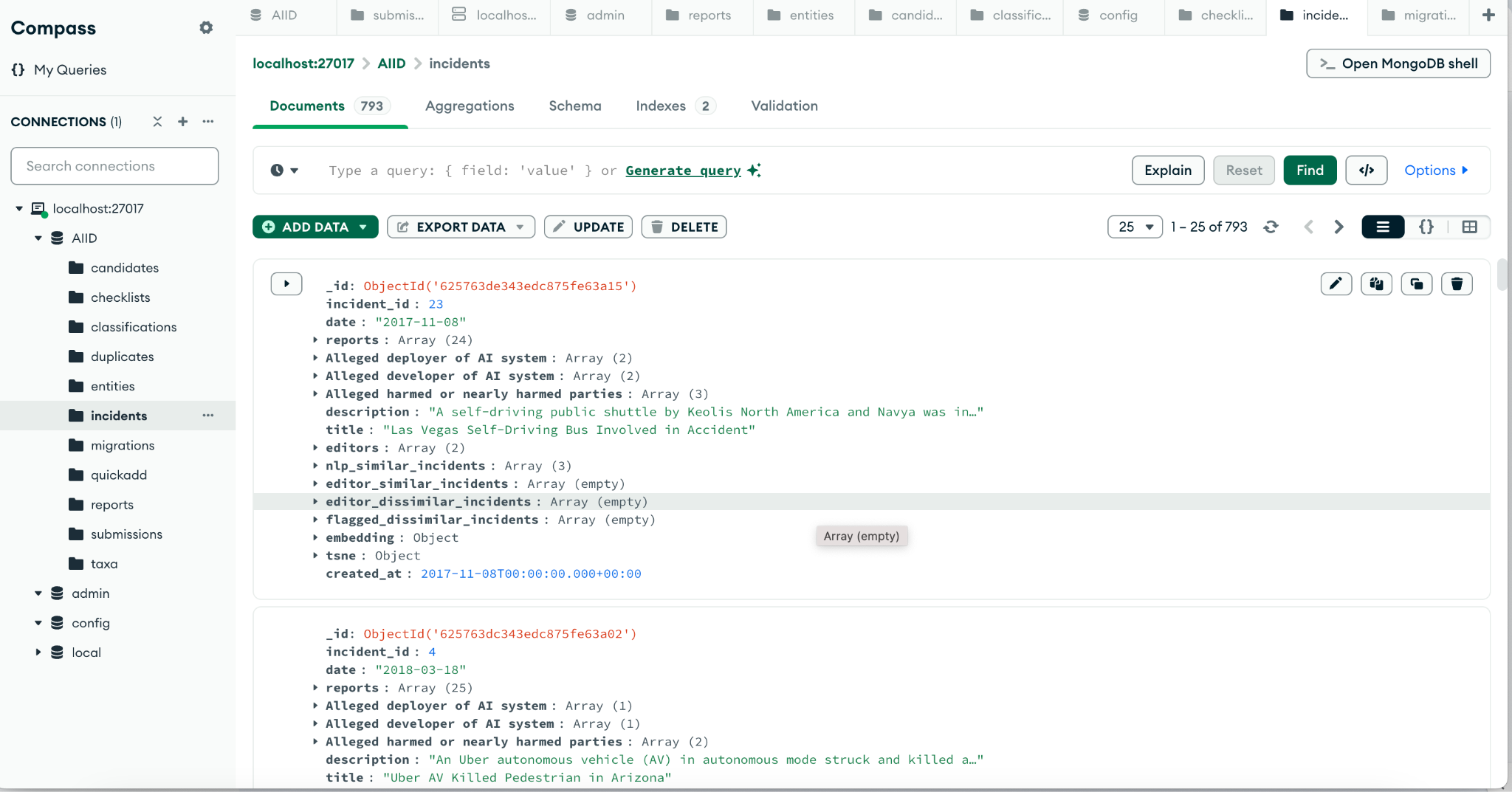
**Computational Bottlenecks:**

Another significant challenge was the computational power during the implementation. The resource-intensive nature of the LLMs, particularly when using large models like GPT-4, required substantial processing power, making scalability a persistent concern. These computational demands posed challenges to running analyses efficiently within the available resources.

**Clustering Issues:**

Clustering also has issues, as some entity clusters lacked interpretability, reducing their overall usefulness. Specific entities that were relevant to particular groups were not represented within the clusters. This limitation affected the practicality of using the clusters for graphing, grouping, and deriving meaningful patterns, underscoring the need for refinement in clustering methodologies. These risks and limitations collectively highlighted areas for improvement in data preparation, model development, and computational strategies.

#### **6****. Appendix**



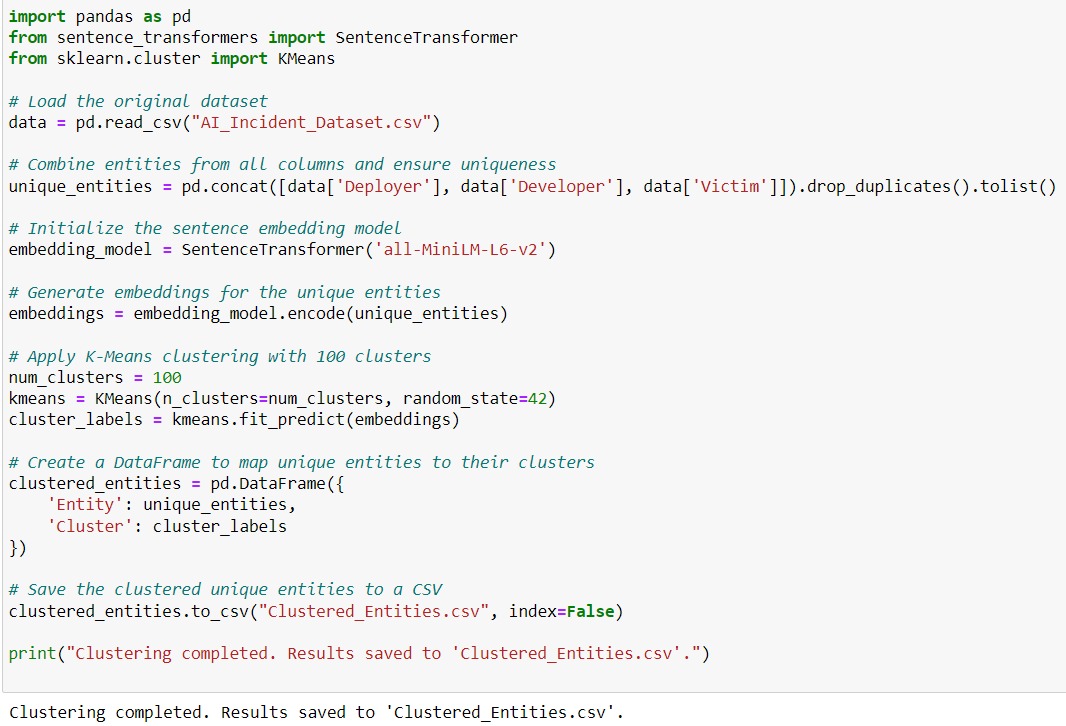
*Fig 5: Snapshot of MongoDB Collections*

**RAG Prompt**

A screenshot of a computer

Description automatically generated

*Fig 6: Code Snippet used for RAG results*



*Fig 7: Code snippet used for K-Means Clustering*