**Scenario**

I modeled a cybersecurity focus on a hospital that just got hit with an earthquake and is in need of disaster rescue response. The goal was to represent how robotic agents operate in risky environments where cyber anomalies can affect their task execution. The graph simulated agents, tasks, zones, sensors, and different types of cyberattacks like sensor or GPS spoofing to explore how knowledge graphs can support trust modeling and anomaly detection in dynamic environments.

**Graph Structure**

I used the Neo4j knowledge graph that includes:

* Agents (robots/drones): RescueBot\_A1 and Drone\_B2
* Tasks: EvacuatePatient, DeliverMeds
* Zones: Room101, CorridorEast
* Sensors: GPS and Camera
* Anomalies: GPS spoofing, severity level high

The relationships for each node include:

* Performs: agent to task
* Operates\_in: agent to zone
* Monitored\_by: zone to sensor
* Compromised\_by: zone to anomaly

This structure reflects the scenario both in operational flow and potential security risks within the system.

**Graph Generator**

The Python script uses NetworkX to generate a larger-scale version of the graph consisting of 9K nodes and 6K edges, with agents randomly connected to zones and tasks, and a percentage of zones compromised by randomly generated anomalies.

**Relevance to ML, GNN/ and Quantum Takss**

The graph focuses on GNN-based anomaly detection such as node or edge classification to detect spoofed or faulty behavior. In cybersecurity research, it modeled adversarial attacks and analyzed failure propagation. In quantum optimization, it focused on routing or task allocation using trust-aware QUBO logic on subgraphs. This relates to my research since it tests environments applying GNNs to detect adversarial patterns and evaluates how cyberattacks impact navigation or coordination in robotic systems.