Securing Dynamic Robotic Behavior in Unpredicted Environments: Enhancing Trust through Adaptive Learning and Cyber Defense

Giselle Roman | Dr. Yugyung Lee | 06/30/2025

Problem Statement Overview

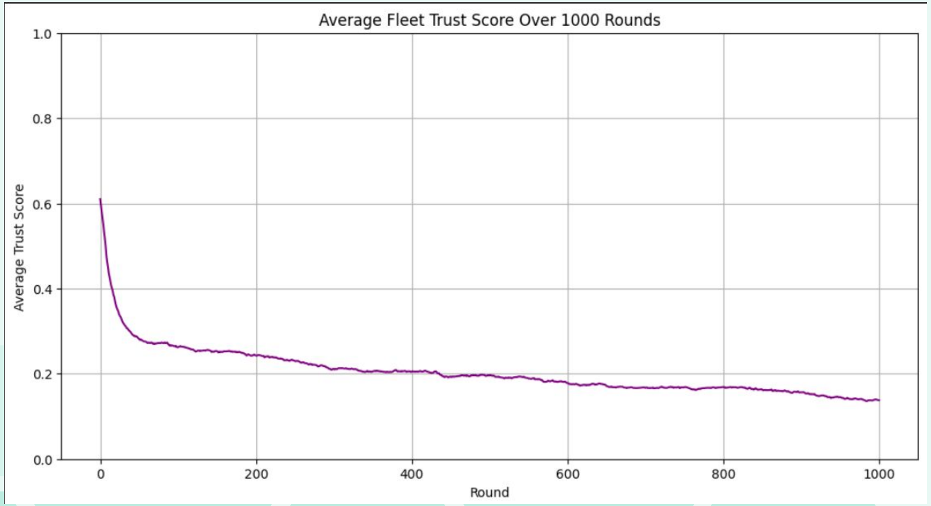
Develop a GNN-based anomaly detection system for securing robotic behavior in post-disaster environments. Detect cyber-physical threats like GPS spoofing and sensor tampering in real-time. Enable adaptive trust-aware task allocation with QUBO optimization for flexible multi-agent coordination.

Paper 1: Graph Poisoning Attacks

- Poisoning attacks degrade embedding quality without direct supervision
- Small, targeted changes can harm downstream tasks
  - Node classification and link prediction
- DeepWalk (unsupervised)
  - Learns node embeddings by random walks on the graph
- Targeted Attacks (misclassification)
  - Degrade test accuracy by 10–20%
- Transferable Attacks
  - General threat to any graph-based system using node embeddings
  - Poisoned graph crafted for DeepWalk also harms node2vec

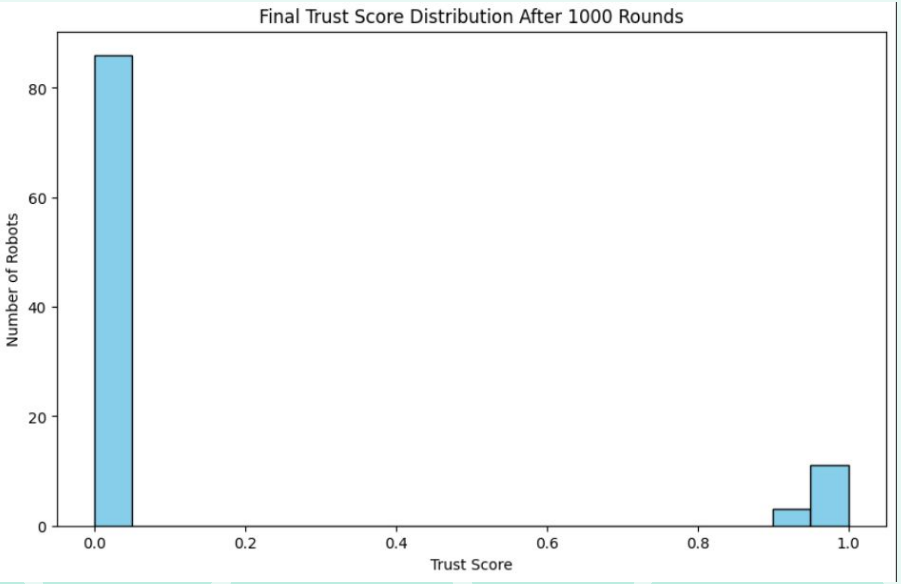
Simulation Results – No Defense

- Shows steady decline in average trust score across the fleet
- Starts ~0.6 but drops below 0.2 by end of 1000 rounds
- Trust erodes over time because ~10% of robots are compromised each round
- No recovery mechanism
  - Trust lost is permanent



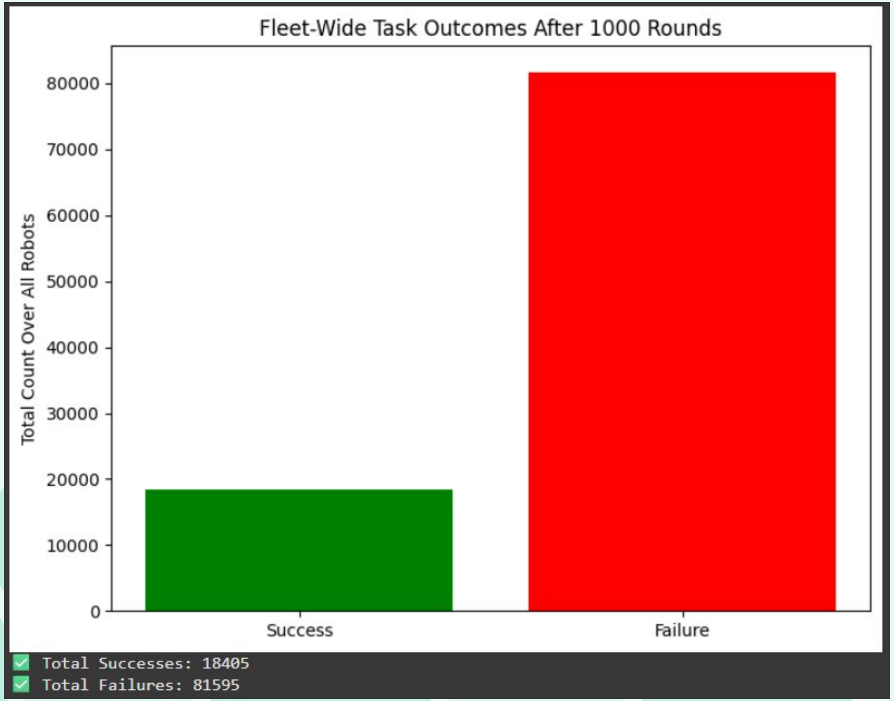
Simulation Results – No Defense

- Most robots end with very low trust scores (near 0)
- Sharp peak at 0.0
- Very few robots maintain high trust (~1.0)
- Widespread compromise with no defense to restore trust



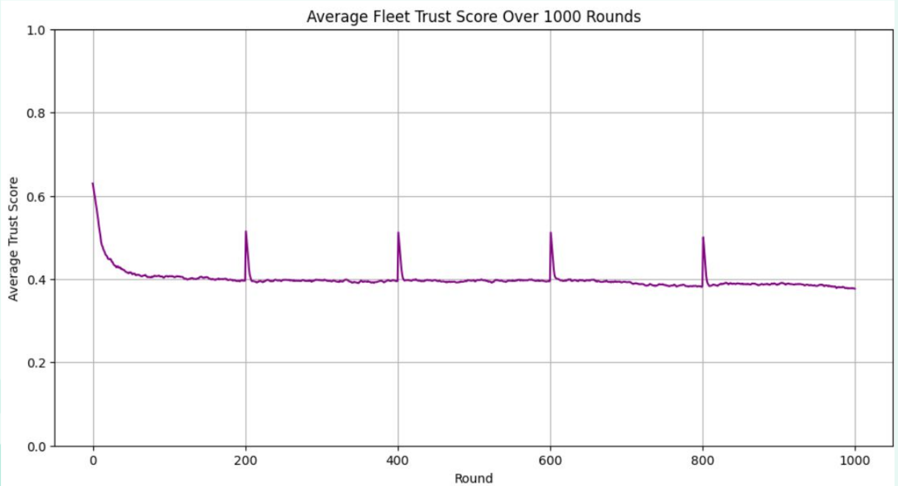
Simulation Results – No Defense

- Comparing successes vs. failures
- Successes: ~18,000 total
- Failures: ~81,000 total (over 4x more)
- High failure rate reflects degraded trust
  - More compromised or unreliable agents failing tasks



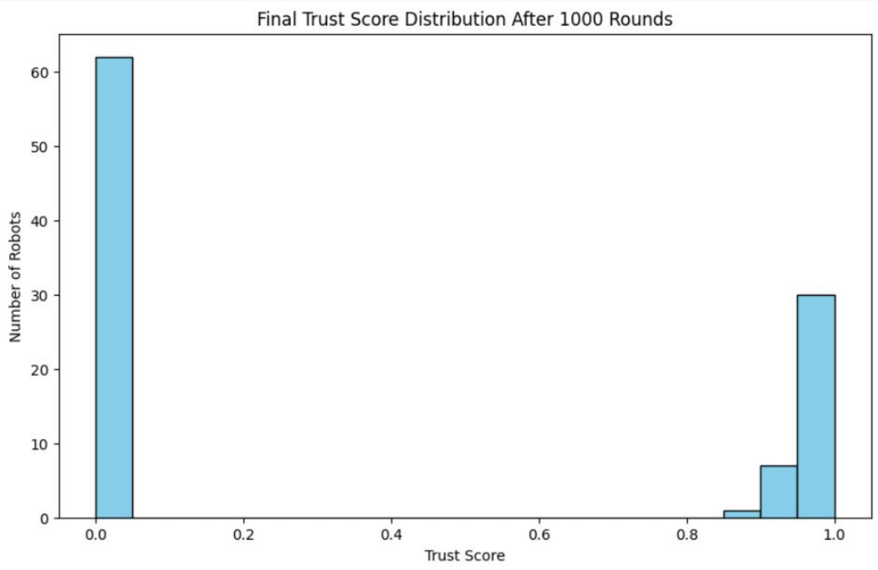
Simulation Results – With Defense

- Average trust stabilizes around ~0.4 over time
- Sharp periodic spikes every ~200 rounds from trust floor countermeasure
- Countermeasures limit degradation despite ongoing attacks
  - Trust restoration mechanisms



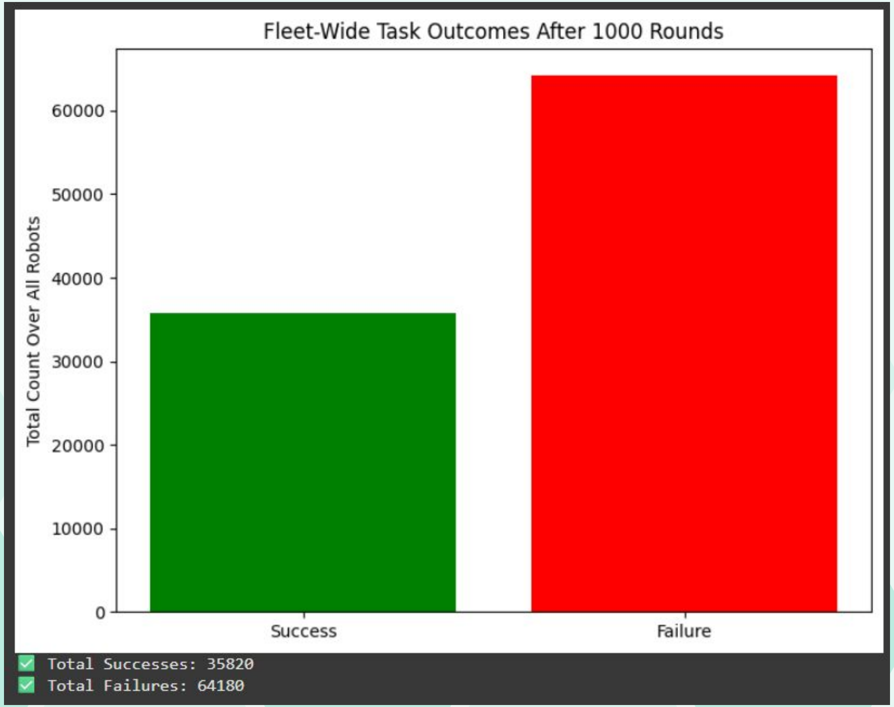
Simulation Results – With Defense

- More robots maintain moderate to high trust scores
- Two peaks
  - Many agents still near 0 (compromised)
  - Significant cluster near ~1.0 (trusted, reliable)
- Defense prevents entire fleet from collapsing to 0 trust



Simulation Results – With Defense

- Successes: ~35,800
- Failures: ~64,100
- Higher success count compared to no-defense scenario
- Overall failure rate significantly reduced
  - Better task reliability under attack



QUBO Assignment Optimization Attempt

- Trust matrix represents how likely agents are to perform tasks well
- Sampled 5 agents × 5 tasks from 100×100 matrix
- Trust values vary from near-zero to ~0.95
- Low rows indicate compromised agents

```
5x5 Trust Matrix for QUBO
[[0.78129562 0.81890281 0.95029172 0.8429687  0.95864335]
 [0.         0.         0.         0.         0.        ]
 [0.80545941 0.8442297  0.97968219 0.86903989 0.98829211]
 [0.00805459 0.0084423  0.00979682 0.0086904  0.00988292]
 [0.00805459 0.0084423  0.00979682 0.0086904  0.00988292]]
```

QUBO Assignment Optimization Attempt

- Shows how adversarial compromise weakens confidence in assignments
- Simulated attack on Agent 2 (3rd row) by reducing trust by 0.4
- Trust scores dropped ~40%, many near-zero

```
Poisoned Trust Matrix
[[0.78129562 0.81890281 0.95029172 0.8429687  0.95864335]
 [0.         0.         0.         0.         0.        ]
 [0.40545941 0.4442297  0.57968219 0.46903989 0.58829211]
 [0.00805459 0.0084423  0.00979682 0.0086904  0.00988292]
 [0.00805459 0.0084423  0.00979682 0.0086904  0.00988292]]
```

QUBO Assignment Optimization Attempt

- Trust restoration countermeasures
- Simple defense
  - Added 0.2 to values < 0.3
- Recovered very low trust regions
- Results in slightly improved trust matrix

```
Defended Trust Matrix
[[0.78129562 0.81890281 0.95029172 0.8429687  0.95864335]
 [0.2        0.2        0.2        0.2        0.2       ]
 [0.40545941 0.4442297  0.57968219 0.46903989 0.58829211]
 [0.20805459 0.2084423  0.20979682 0.2086904  0.20988292]
 [0.20805459 0.2084423  0.20979682 0.2086904  0.20988292]]
```

QUBO Assignment Optimization Attempt

- Ran QUBO solver on Poisoned and Defended matrices
- Poisoned Assignment
  - Top row: many 1’s
    - Over-assignment
  - Middle rows:
    - under-assignment or sparse
- Shows degraded coordination under attack

```
QUBO Assignment (Poisoned)
[[1 1 1 1 1]
 [0 0 0 0 0]
 [0 0 1 0 1]
 [0 0 0 0 0]
 [0 0 0 0 0]]
```

QUBO Assignment Optimization Attempt

- Defended Assignment
  - Slightly improved sparsity
  - Still not perfect 1-to-1 mapping
  - Some rows remain all zeros
- Even with defense, simple QUBO formulation fails to enforce one-task-per-agent perfectly
- Penalties were too soft or trust matrix still too degraded

```
QUBO Assignment (Defended)
[[1 1 1 1 1]
 [0 0 0 0 0]
 [0 0 1 0 1]
 [0 0 0 0 0]
 [0 0 0 0 0]]
```

Challenges & Limitations

- Full 100×100 trust matrix too large for exact QUBO solver
- QUBO assignments often invalid
  - Over-assignment or under-assignment
  - Constraints sometimes not fully enforced despite penalty terms
- Simple defense only partially improved trust
- Defended matrices still produced invalid assignments in some runs
- Need better scaling methods for large deployments

Next Steps

- Refine QUBO formulation and constraint handling
- Explore alternative solvers or heuristics for scaling
- Test with larger agent/task samples
- Incorporate GNN-based anomaly detection outputs into trust simulation
- Improve defense strategies for adversarial robustness

References

- Bojchevski, Aleksandar, and Stephan Günnemann. “Adversarial Attacks on Node Embeddings via Graph Poisoning.” ArXiv.org, 2018, arxiv.org/abs/1809.01093.