Assignment, WASP Software Engineering Course Module 2025

Yinuo Zhang
yinuo.zhang@cs.umu.se

1. Introduction

My PhD work is about cybersecurity in the Cloud-Edge Continuum, which is starting to form the backbone of systems in areas like finance, healthcare, and smart homes. At a basic level, it's a network made up of a central node and lots of smaller edge nodes, all talking to each other constantly.

In a traditional cloud setup, that central node does nearly all the heavy lifting—processing data, storing it, and training models. That works fine, but for users far away, it can mean delays. The Cloud-Edge idea tackles this by pushing computing power closer to where people are, which speeds things up. Of course, there's a trade-off: more connected nodes also mean more opportunities for attackers. Attacks like Distributed Denial of Service (DDoS), Man-in-the-Middle (MitM), and malware injection don't need to be reinvented to cause trouble here—they just need to be adapted to the new setup.

Currently, I'm focusing on detection, with DDoS as my example case. These attacks, around since the late 1990s, basically flood a system so real users can't get through. Detection is usually treated as a classification problem: spot the bad traffic quickly and accurately so it can be blocked. Interestingly, recent research shows that Large Language Models (LLMs)—built for things like predicting the next word in a sentence—also pick up other skills along the way, like sentiment analysis and translation. Those emergent properties could help recognise abnormal traffic patterns [1].

The big hurdle for me is efficiency. Edge nodes don't have the power or storage for giant models like GPT-3 [2]. My aim is to design a smaller, optimised LLM that can be trained on powerful central servers but still run well on the edge. The challenge is to keep performance high while keeping the size and resource demands low [3]. If I can pull that off, it means the Cloud-Edge Continuum can stay fast and accessible without opening the door wider to cyberattacks.

2. Lecture Principles

2.1 Principle 1: cognitive bias mitigation in cybersecurity model development

From the lecture, I learned that cognitive biases—like confirmation bias, availability bias, and anchoring—affect not only human decision-making but also how software systems are developed and evaluated. In my research, these biases could influence how DDoS detection models for the Cloud-Edge Continuum are designed and validated. For instance, confirmation bias might push me toward datasets

that match expected attack patterns, while availability bias could make me over-focus on recent, well-known attacks.

Mitigating this requires deliberate strategies, such as testing against traffic that mimics benign patterns or exploring multiple detection approaches before deciding on one. In the Cloud-Edge setting, where traffic can be highly region-specific, guarding against these biases is essential to avoid overfitting and ensure robust security across the network.

2.2 Principle 2: prioritising validation in ML-based security systems

The Quality Assurance lecture highlighted the difference between verification ("are we building the product right?") and validation ("are we building the right product?"), noting that in machine learning, validation often matters more. In my research, a DDoS detection model could score well on accuracy or precision yet still fail if it cannot handle rare traffic patterns or adapt to evolving attacks.

Validation means testing whether the model truly meets Cloud-Edge operational needs: detecting low-volume, slow-burn attacks, working under resource limits, and keeping high recall without flooding the system with false positives. This also requires diverse and representative test data to uncover weaknesses, such as performance drops in certain time windows or regions. Embedding these checks ensures my optimised LLM-based detection system is effective not just in the lab but in the complex, variable conditions of the Cloud-Edge Continuum [4].

3. Guest-Lecture Principles

3.1 Principle 1: problem-space vs. solution-space thinking

The first concept from the guest lecture that aligns with my research is the distinction between problem-space and solution-space thinking. In Cloud-Edge cybersecurity, it's tempting to jump straight into building a detection model—like an optimised LLM for edge nodes—without first clarifying the real operational need. Staying in the problem-space helps me define the core challenge: detecting evolving DDoS attacks in diverse, resource-limited edge environments [5]. Once that is clear, I can move to the solution-space and compare different approaches—LLM-based, lightweight ML, or hybrid models—against those requirements. This separation helps avoid committing too early to a solution that may fail under real-world conditions.

3.2 Principle 2: goal modelling and goal refinement

The second concept is goal modelling and goal refinement. This links high-level aims (e.g., "detect and block malicious traffic in real time") with system constraints ("run within 200 MB memory, 50 ms

inference") and business priorities ("99.99% uptime"). Mapping and refining these goals highlights trade-offs—such as accuracy versus resource use—and keeps solutions aligned with operational realities. In the Cloud-Edge setting, where constraints and threats differ across nodes, this structured alignment is essential for delivering effective, consistent protection.

4. Data Scientists versus Software Engineers

4.1 Answer to question 1: Agreement with the CMU distinctions

Yes, I agree with the CMU book's description of the differences between data scientists and software engineers. Data scientists often work in an exploratory and experimental mode—testing models, cleaning data, and iterating quickly—while software engineers focus more on building reliable, maintainable, and scalable systems. These approaches are complementary but can create gaps when a model moves from a notebook into production. In my own research, I have seen how this divide can slow down deployment and reduce overall system quality.

4.2 Answer to question 2: The need for Cross-Skilling

I believe the future will require both roles to learn skills from the other side. In production ML systems—especially in complex environments like the Cloud-Edge Continuum—a model cannot succeed without considering production constraints, and a system cannot be robust without understanding model behaviour. Data scientists need some engineering discipline for deployment, monitoring, and optimisation. Engineers need enough ML knowledge to handle model drift, interpret performance metrics, and adapt to changing data. As these skills overlap, the boundary between the roles will blur, leading to more integrated and effective teams.

5. Paper analysis

5.1 Paper 1: Modeling Resilience of Collaborative AI Systems

1) Core Idea & SE Importance

The paper sets out a way to systematically measure resilience in Collaborative AI Systems (CAIS) by tracking system states (steady, disruptive, recovered) and defining metrics such as the Autonomous Classification Ratio (ACR), state durations, and human intervention rates [6]. It is important because it transforms resilience from an informal "we think the system bounces back" into something measurable and comparable between systems.

2) Relation to My Research

In my Cloud-Edge Continuum DDoS detection work, disruptions happen in the form of sudden malicious traffic spikes, partial outages, or even degraded bandwidth. Having a standard way to describe and measure these events is directly relevant.

3) Integration into a Larger AI Project

A large-scale, edge-based intrusion detection platform could use this approach to monitor how quickly different nodes recover after an attack [7] and whether they need manual intervention.

4) Adapting My Research

Right now, my research goals are heavily performance-focused: higher detection accuracy, smaller model size, lower latency. But I haven't built resilience tracking into the models or the infrastructure. The framework in this paper changes how I think about the whole problem.

I would begin by defining resilience metrics that make sense for a DDoS detection context. For example:

- Detection ACR: proportion of packets or flows correctly classified without human help, especially under load.
- Mean Disruption Duration: how long the system remains in a degraded state after the onset of an attack.
- Recovery Efficiency: ratio of performance regained to resources spent in recovery.

To make these useful, each edge node would need to monitor its own state in real time. This could be done by embedding a state-tracking module that labels periods as "steady," "under attack," and "recovered." Those labels would be determined by a combination of detection accuracy trends, latency, and packet drop rate. All of this data would be reported to a centralized dashboard, where operators could spot patterns: for example, certain geographic regions recovering slower, or specific hardware configurations being more resilient.

One immediate change to my research workflow would be introducing stress testing as a core part of evaluation. Right now, like many in ML, I evaluate on static datasets. But in a resilience-focused approach, I would simulate bursts of attack traffic, random packet loss, or sudden shifts in legitimate traffic patterns, and measure not only how detection accuracy changes but how quickly it returns to baseline. This kind of testing might even reveal that some "high-accuracy" models are brittle under real-world conditions.

I would also develop fallback detection models. These would be smaller, rule-based or distilled LLMs that can be deployed instantly when the main model performance drops. The goal would be to avoid total service loss: even if the fallback is less accurate, it maintains a safety net for the system while the main

model recovers or retrains. Deciding when to trigger a fallback could be tied directly to the resilience metrics—e.g., if ACR drops below 70% for more than a few seconds, switch models.

Long term, this approach could produce a resilience profile for every model I build. Just as we compare models based on accuracy, F1-score, and latency, we could add recovery time and robustness under disruption as standard evaluation dimensions. This would mean my research outputs aren't just "fast and accurate" in a lab setting—they're also reliable in the unpredictable, adversarial conditions of real-world cloud-edge deployments.

5.2 Paper 2: GResilience: Trading Off Between Greenness and Resilience of Collaborative AI Systems

1) Core Idea & SE Importance

The paper expands the resilience framework by factoring in energy consumption [8]. It points out that recovery actions—especially fast, robust ones—often require more energy, and in constrained environments that can be a problem. By framing the choice between resilience and "greenness" as an optimisation problem, it gives AI engineers a structured way to make trade-offs that suit their deployment context.

2) Relation to My Research

My optimised LLM DDoS detection models already aim to be efficient, but energy usage isn't currently a measured or optimised variable—especially during recovery. That's a gap this paper highlights.

3) Integration into a Larger AI Project

In a global edge intrusion detection system, some nodes will run on battery or renewable energy. Recovery policies that ignore energy availability could drain those nodes [9], taking them offline entirely. A GResilience-inspired decision layer would avoid that.

4) Adapting My Research

To align with the GResilience approach, I would need to treat energy use as a first-class performance metric in my research. That means:

- Energy Profiling for All Model Variants: I'd measure the energy cost of the full LLM, a mid-size distilled version, and a minimal fallback model under realistic network traffic loads. These measurements would be taken not just in idle conditions but during high-load recovery scenarios.
- Dynamic Model Switching: I would build a runtime controller that decides, based on both threat severity and available energy, which model to run. If the system is on mains power and sees a serious threat, it can afford to load the largest model. If it's on battery and the threat level is moderate, it should stick with a smaller one.

- Recovery-Energy Trade-Off Modelling: Using either optimisation algorithms or reinforcement learning, I'd create a decision policy that explicitly weighs resilience gains against energy cost. The goal would be to maximise "resilience per watt," a metric that could guide real-time decisions.
- Energy-Aware Stress Testing: Similar to the resilience tests in Paper 1, but with an added constraint: simulate attacks under varying energy budgets. This would reveal if the system can maintain acceptable detection performance without exhausting resources.

One of the more interesting implications is that this might lead me to design multi-stage recovery strategies. For example:

- On detecting a performance drop, first try a lightweight, low-energy intervention—maybe refreshing a subset of the model parameters.
- If that fails and energy is sufficient, escalate to loading a larger model or performing a full retrain.
- If energy is limited, stay in "containment mode" until conditions improve.

Adopting this philosophy means my research would no longer optimize models for accuracy in isolation. Instead, every model would be characterised by a three-way balance: detection accuracy, resilience under disruption, and energy efficiency during recovery. This would produce detection systems that are genuinely fit for the realities of the Cloud-Edge Continuum—where uptime, resource limits, and sustainability all matter at once.

6. Research Ethics & Synthesis Reflection

1) Search and Screening Process

I went through the CAIN conference site and focused on the latest proceedings (2023–2024), only looking at full papers. I searched for keywords like resilience, collaborative, energy, edge, and security. After skimming abstracts, I picked two that clearly tied into distributed AI and the kinds of challenges I work on.

2) Pitfalls and Mitigations

A few titles were a bit misleading—some "resilience" papers were really about team management, not system performance. To avoid that, I jumped straight to the methods section before committing. I also widened my search terms from "edge security" to "distributed AI," which gave me better options.

3) Ethical Considerations

I read both papers myself, made my own notes, and wrote everything in my own words, no copy-pasting from the papers or AI tools, though I did use an AI assistant to help refine the final wording

References

[1] Ferrag, Mohamed Amine, Fatima Alwahedi, Ammar Battah, Bilel Cherif, Abdechakour Mechri, Norbert Tihanyi, Tamas Bisztray, and Merouane Debbah. "Generative ai in cybersecurity: A comprehensive review of llm applications and vulnerabilities." Internet of Things and Cyber-Physical Systems (2025).

[2] Moitra, Abhishek, Abhiroop Bhattacharjee, Youngeun Kim, and Priyadarshini Panda. "RobustEdge: Low power adversarial detection for cloud-edge systems." IEEE Transactions on Emerging Topics in Computational Intelligence 8, no. 2 (2024): 2101-2111.

[3] Zhong, Duo, Bojing Li, Xiang Chen, and Chenchen Liu. "EdgeShield: A Universal and Efficient Edge Computing Framework for Robust AI." arXiv preprint arXiv:2408.04181 (2024).

[4] Rahmati, Milad. "Towards Explainable and Lightweight AI for Real-Time Cyber Threat Hunting in Edge Networks." arXiv preprint arXiv:2504.16118 (2025).

[5] Ma, Yong, Long Liu, Zhiquan Liu, Fagen Li, Qilin Xie, Kaiwei Chen, Chenyang Lv, Ying He, and Fan Li. "A survey of ddos attack and defense technologies in multi-access edge computing." IEEE Internet of Things Journal (2024).

[6] Rimawi, Diaeddin, Antonio Liotta, Marco Todescato, and Barbara Russo. "Modeling Resilience of Collaborative AI Systems." In Proceedings of the IEEE/ACM 3rd International Conference on AI Engineering-Software Engineering for AI, pp. 24-29. 2024.

[7] Li, Mingyue, Liudong Zheng, Xiaoxue Ma, and Shuang Li. "Real-time monitoring model of DDoS attacks using distance thresholds in Edge cooperation networks." Journal of Information Security and Applications 89 (2025): 103972.

[8] Rimawi, Diaeddin, Antonio Liotta, Marco Todescato, and Barbara Russo. "GResilience: trading off between the greenness and the resilience of collaborative AI systems." In IFIP International Conference on Testing Software and Systems, pp. 266-273. Cham: Springer Nature Switzerland, 2023.

[9] Shaffi, Shamnad Mohamed, Sunish Vengathattil, Jezeena Nikarthil Sidhick, and Resmi Vijayan. "AI-Driven Security in Cloud Computing: Enhancing Threat Detection, Automated Response, and Cyber Resilience." arXiv preprint arXiv:2505.03945 (2025).