WASP | WALLENBERG AI, AUTONOMOUS SYSTEMS AND SOFTWARE PROGRAM

Threat Modelling for AI-Based Radio Access Networks (AI-RAN): Bridging the Gap Between Theory and Practice

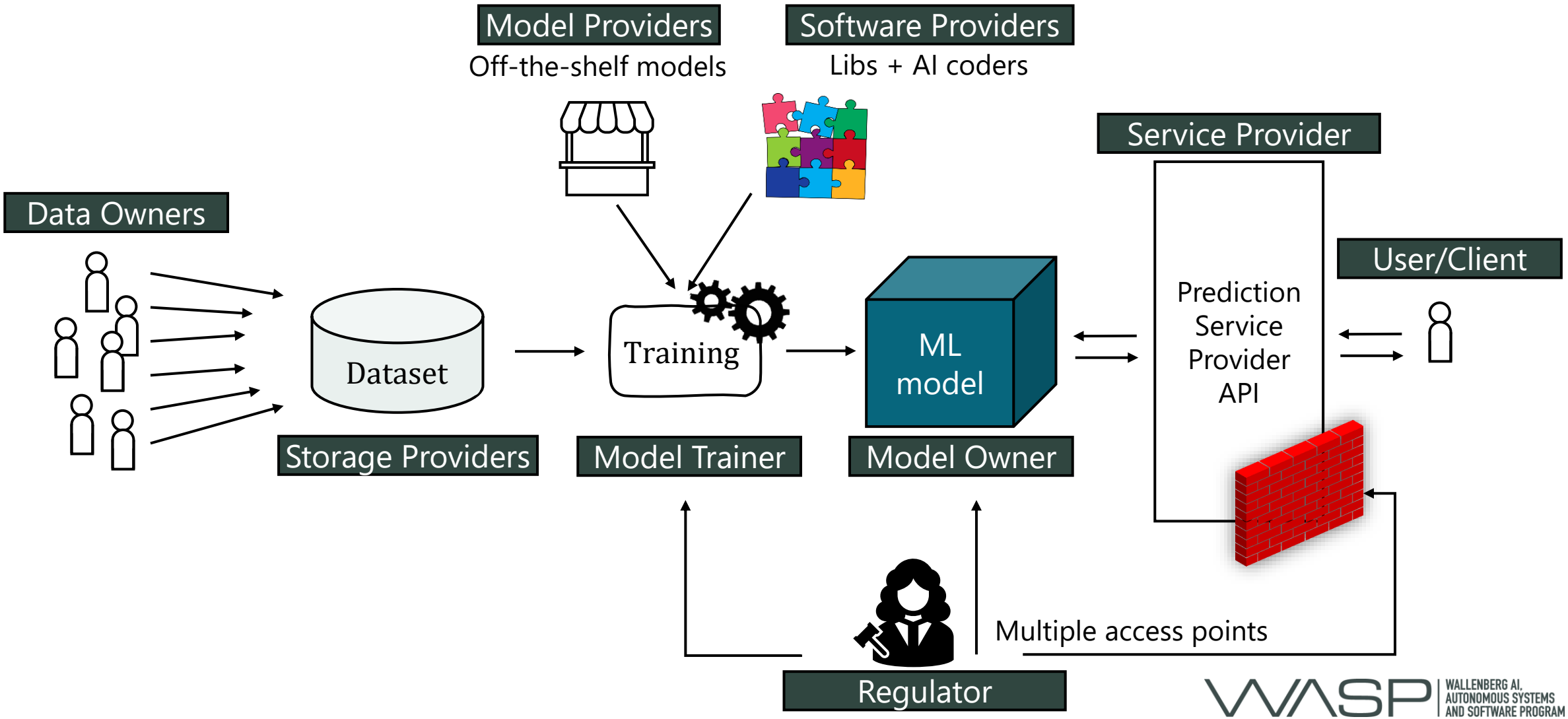
Buse Atli

System Security Workshop, April 2026

Thanks to Antti Konttinen, Esa Metsala, Mehrnoosh Monshizadeh

https://busegulatli.github.io/
https://liu.se/en/employee/busat07

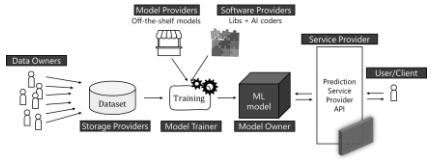
Machine Learning (ML) Pipeline



WASP WALLENBERG AI, AUTONOMOUS SYSTEMS AND SOFTWARE PROGRAM

Threat Modelling Framework

System & Assets
What are we working on?



- Model integrity in training/test time
- Input data privacy
- Model privacy
- User privacy
- Decisions
- Robustness
- Safety

Threats
What can go wrong?

- Adversarial inputs at test time (model evasion)
- Data & model poisoning
- Abuse of API & Denial of service
- Model extraction
- Dataset inference
- Attribute inference
- Membership inference

Attacker Model
Who might do it and how?

- **Who:** End user, data provider, model trainer, software provider
- **Access:** Black-box (input queries + outputs) vs. white-box
- **Capabilities:** input control, repeated interaction, training code, initial model weights

Defenses & Evaluation
What we are going to do and how do we know it works?

- **Mitigations:** Robust training methods, detection of attacker, input validation, API monitoring, verifiable mechanisms
- **Evaluation:** Attack simulations/red teaming
- **Metrics:** e.g., certificates, attack success rate etc.
- Continuous assessment
- Feedback from **regulator**

WASP | WALLENBERG AI, AUTONOMOUS SYSTEMS AND SOFTWARE PROGRAM

Model Evasion I

Attacker profile:

- (Untrusted) end user
- **Primary goal:** evasion (incorrect prediction) at inference time
- **Secondary goal:** stealth, minimum # of queries

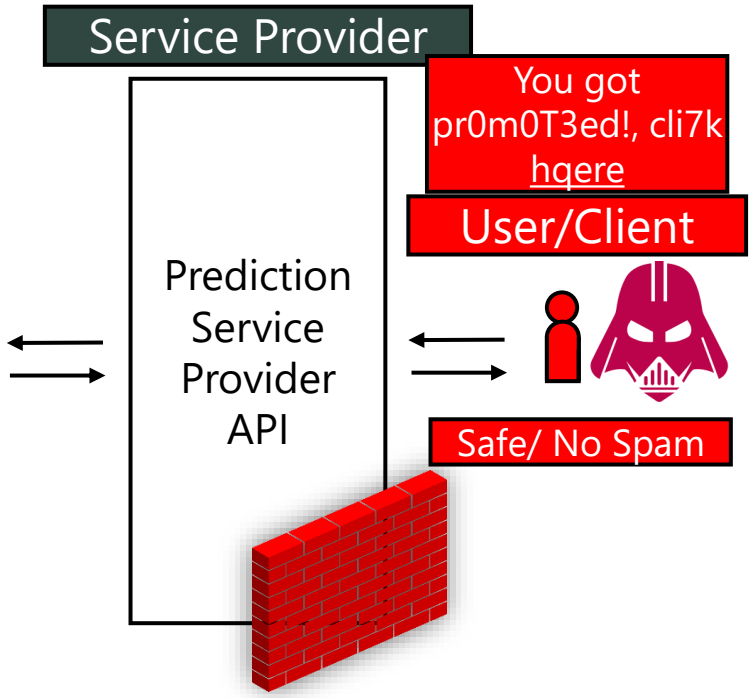
Capabilities and constraints:

- Can submit arbitrary inputs
- Can observe outputs: (top-k) labels, confidence scores
- Can issue multiple queries
- May have black-box, partial knowledge (grey-box) or white-box
- Rate-limited user accounts
- Perturbation bound and imperceptibility

Attack surface:

- Input surface: raw features (pixels, tabular values), preprocessing pipeline
- Model interface: prediction API and outputs

$$x' = x + \delta \;\; s.t. \; f(x')! = f(x)$$



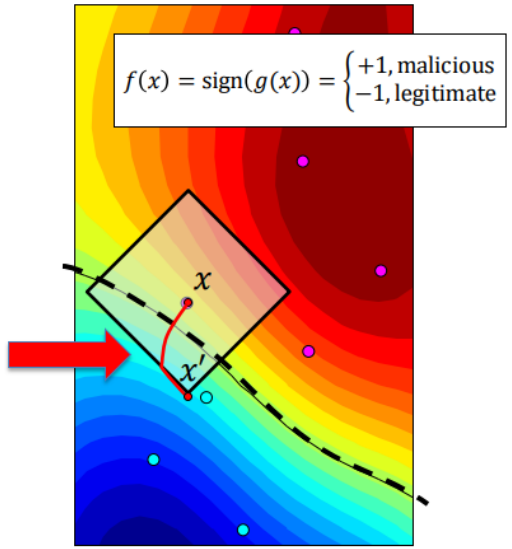
Model Evasion II

Impact: Security bypass e.g., malwares, safety issues in autonomous systems

Core idea: Models rely on **brittle decision boundaries** that can be crossed with minimal change on inputs

Defenses:

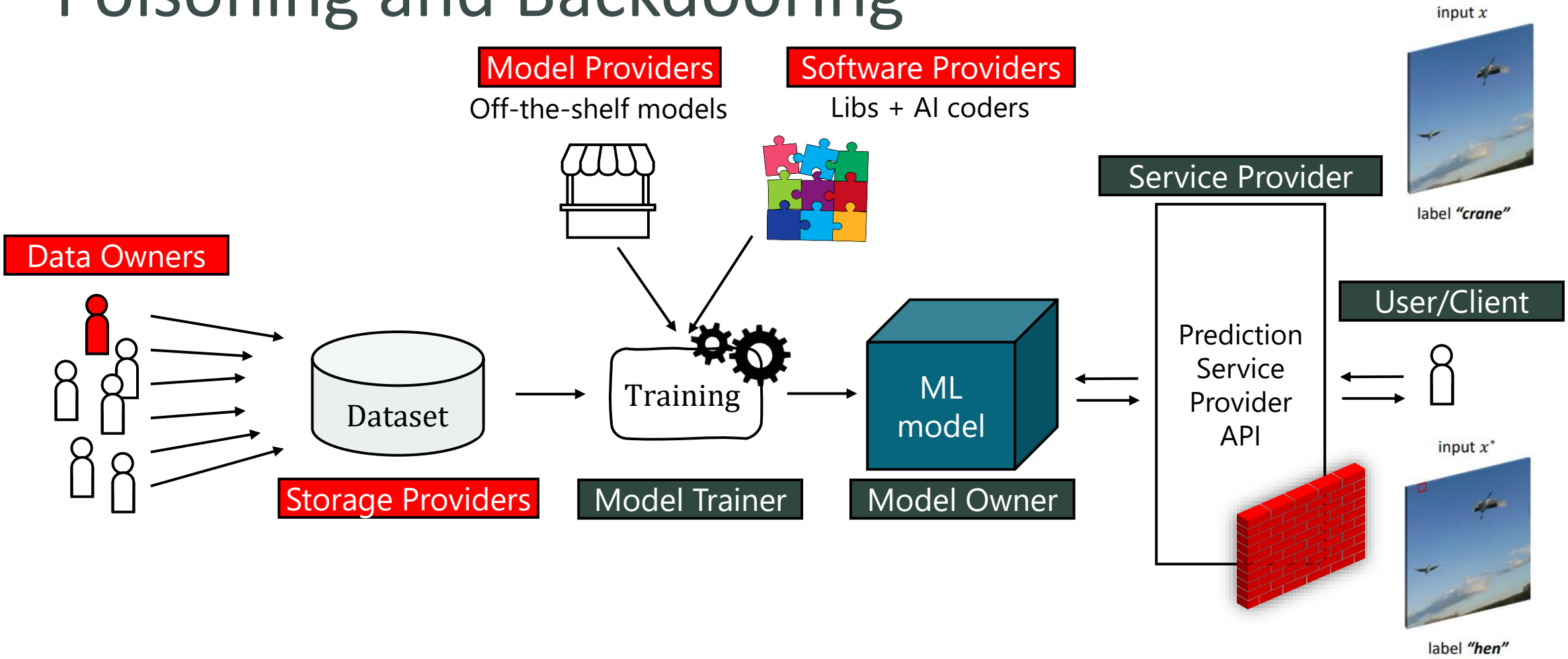
- **System level:** Limit output granularity, rate-limiting, API monitoring
- **Detection-based:** Detecting attacker's queries
- **Preprocessing:** Compression, random resizing, denoising
- **Adversarial training:** Including adversarial examples into the training phase
- **Certified robustness:** Provable guarantees that the model can resist all adversarial perturbations within a specified threat model and perturbation bound



$$\mathbb{B}_p(\mathbf{x}_0, \epsilon) = \{\mathbf{x} \,|\, ||\mathbf{x} - \mathbf{x}_0||_p \leq \epsilon\}$$

https://unica-mlsec.github.io/mlsec/slides/03-Evasion-Attacks.pdf

Poisoning and Backdooring



Shafahiet al. *Poison Frogs! Targeted Clean-Label Poisoning Attacks on Neural Networks.* (NeurIPS, 2018)
Bagdasaryan and Shmatikov. *Blind Backdoors in Deep Learning Models.* (USENIX, 2021)
Carlini et al. *Poisoning and Backdooring Contrastive Learning,* (ICLR, 2022)
Zhang et al. *Persistent Pre-training Poisoning of LLMs.* (ICLR, 2025)
Langford et al. *Architectural Neural Backdoors from First Principles.* (IEEE S&P, 2025)

Data Poisoning and Backdooring I

Attacker profile:

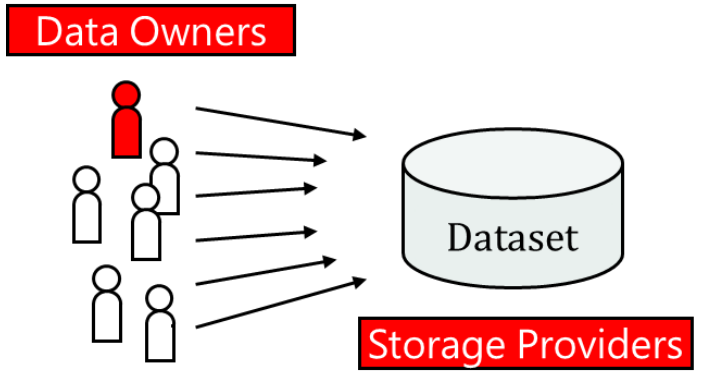
- Malicious data provider
- **Goal:** (Targeted) misprediction on chosen inputs, degrade overall performance

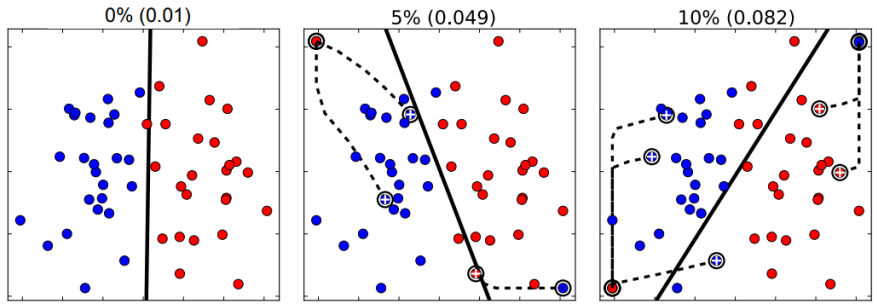
Capabilities and constraints:

- Can add or modify samples to training data, design triggers
- Can modify labels and/or craft clean-label poisons
- Knowledge of data distribution, pre-processing, training pipeline
- Gray-box/black-box model access
- Limited/no access at test time
- Limited # of poisoned samples (e.g., <1%) and no control over full training set

Attack Surface:

- Training data collection, aggregation or storage pipelines
- Attacker is present in pre-training or fine-tuning





Biggio and Roli. Wild Patterns: *Ten Years After the Rise of Adversarial Machine Learning.* (CCS, 2018)
Bagdasaryan and Shmatikov. *Blind Backdoors in Deep Learning Models.* (USENIX, 2021)

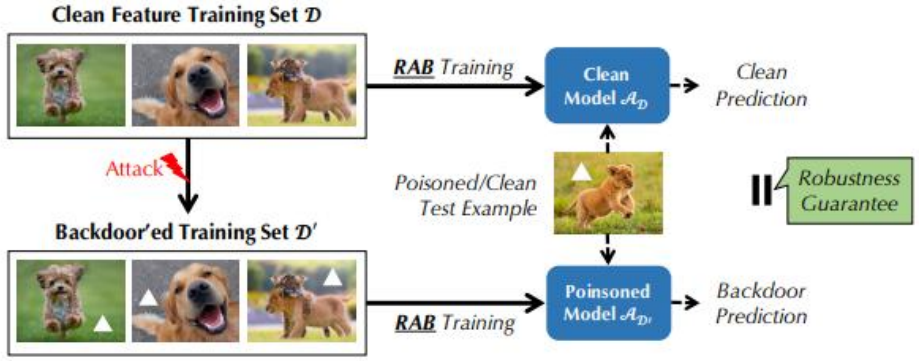
Data Poisoning and Backdooring II

Impact: Integrity violations, persistent and hard to detect failures after deployment, corrupted data pipelines

Core idea: Adversaries can embed **hidden behaviours** into the model that are hard to detect, persist through training, and activate under **specific conditions**

Defenses:

- **System level:** Control and verify data sources, restrict and audit data collection pipelines, dataset versioning and rollback
- **Detection-based:** Outlier or anomaly detection, filtering, model inspection, backdoor reverse engineering
- **Preprocessing:** Strong data augmentations, ensembles, majority voting
- **Robust training:** Fine-pruning, robust loss functions, model cleansing, differential privacy
- **Certified defenses:** Provable robustness using randomized smoothing



Weber et al. *RAB: Provable Robustness Against Backdoor Attacks.* (S&P, 2023)

ML Use Cases for RAN

- **Network Energy Saving:** Reducing energy consumption by switching off underutilized resources
  - Predict traffic/load per cell
  - Identify low-utilization cells and offload traffic to neighbour cells, deactivate underutilized cells
- **Load Balancing:** Avoiding congestion by redistributing traffic across cells
  - Predict cell congestion and identify user equipments (UEs) that can be moved with minimal QoS impact
  - Redistribute traffic across cells to avoid overload
- **Mobility Optimization:** Improving mobility performance with proactive handover decisions
  - Predict UE trajectory
  - Select optimal target cell and optimize handover timing without breaking connection

https://www.cisco.com/c/en/us/products/collateral/wireless/nb-06-radio-access-networks-cte-en.html

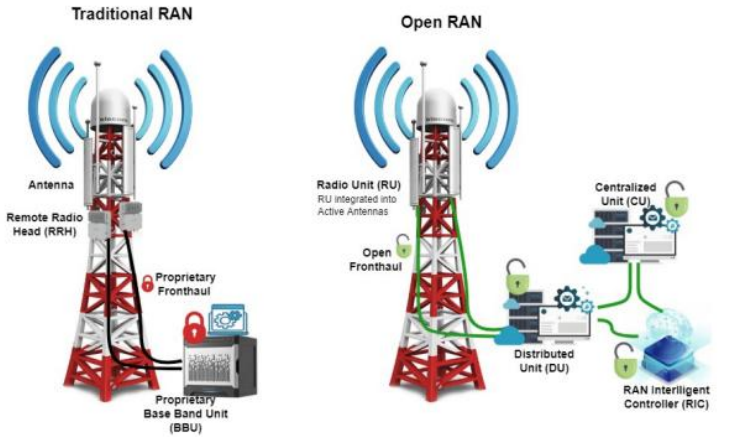
Typical inputs: radio and physical layer measurements, traffic and load-related KPIs, user behaviour data, network configuration and topology data

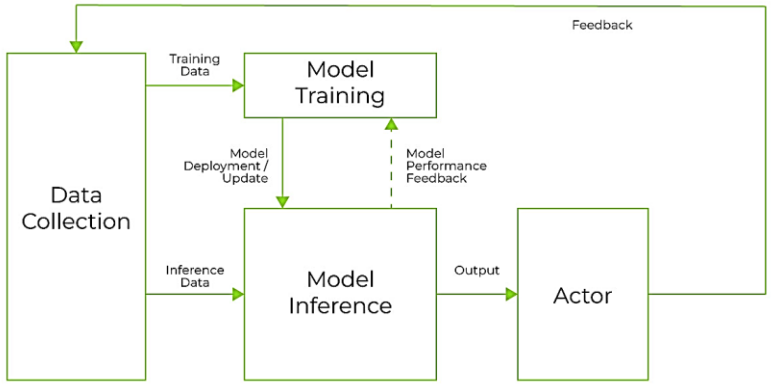
Typical ML models: Time Series Models (LSTM, Transformers) and Reinforcement Learning

WASP | WALLENBERG AI, AUTONOMOUS SYSTEMS AND SOFTWARE PROGRAM

ML in Traditional RAN

- In traditional RAN, base station is treated as a **secure environment** by 3GPP standards
- Use cases include training/inference in base station (BTS), no need for an external system
- Training engine, inference engine and ML data engine (used for dataset collection) are **inside BTS**
  - Training data can be collected from the BTS
  - ML models are not very large or complicated
  - Inference model can be deployed in the BTS
- Management of ML applications and ML models are in Operations Administration and Maintenance (OAM) Management Service and Baseband Unit (BB)



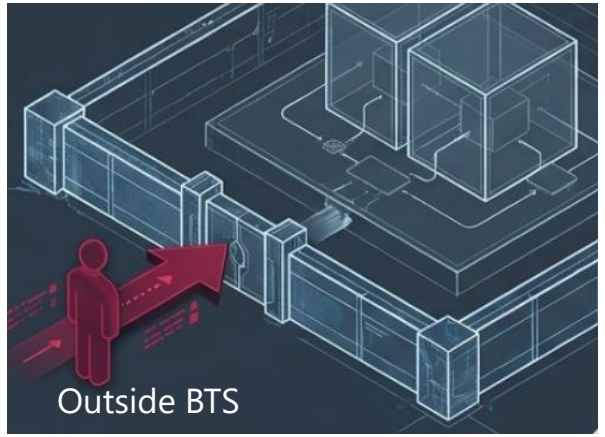


https://www.3gpp.org/technologies/ai-ml-ngran

Liyanage at al. *Open RAN security: Challenges and opportunities.* (Journal of Network and Computer Applications, 2023)

ML Model Management in BTS

- BTS is treated as a **secure environment (trusted boundary)**
  - Internal components (ML models, data, inference) are accessible only to authorized entities and protected from external attackers
- **No integrity protection in** Non-IP Data Delivery (NIDD) scenarios. (e.g., IoT data with minimal headers), **model training configuration file** and/or the trained model file
- **Outside BTS, 3rd party**
  - **Before deployment:** Modify model files or training configuration
  - **Before ingestion:** Inject or alter data in the data collection pipeline
  - **At data sources:** Compromise sensors / UEs / data providers
- **Inside BTS**, development and maintenance teams has direct access to ML data collection engine and internal ML processing

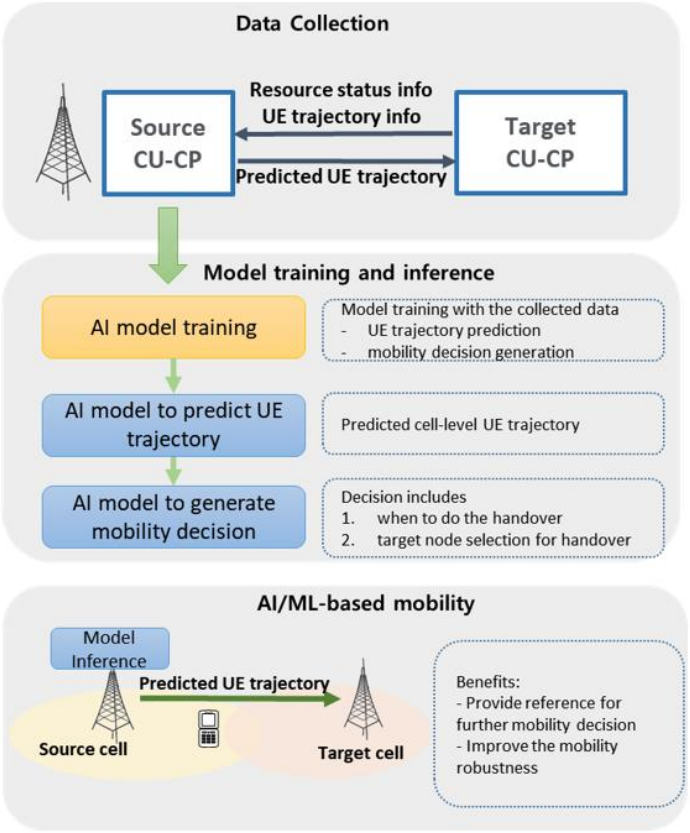


Images created by NotebookLM

Threat Modelling in ML Use Cases

In RAN context:

- **Dataset** -> radio KPIs, UE measurements
- **API** -> OAM / internal interfaces
- **User**, network operator / automated control loop
- Use cases directly translate into **high-impact attacks**
  - Energy saving -> Network outage risk
  - Load balancing -> congestion manipulation
  - Mobility -> handover attacks
- **Model evasion** can cause wrong QoS degradation, load imbalance, or wrong handover decisions
- **Data poisoning** can cause long-term degradation of ML models, biased traffic prediction, and hidden backdoors in network behaviour



https://research.samsung.com/blog/First-Light-of-the-AI-ML-Empowered-RAN-in-3GPP

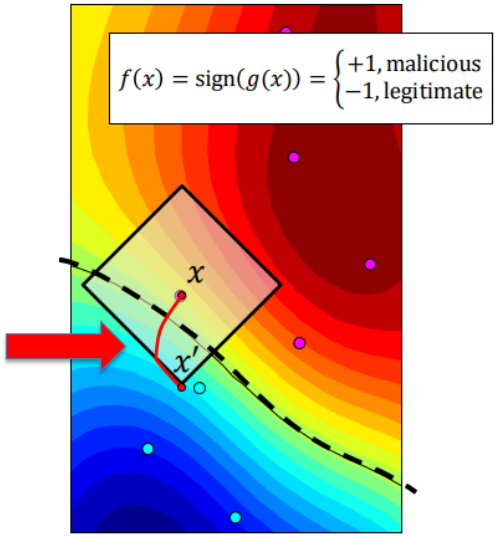
Model Evasion in ML-RAN I

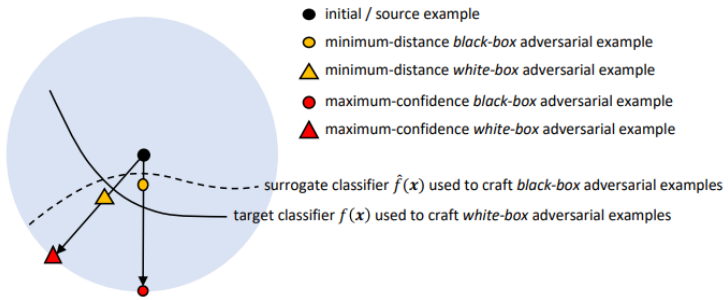
Attacker profile:

- Outside BTS (e.g., compromised UE) and inside BTS (compromised insider actor, e.g., development or maintenance engineer)
- **Example goal:** Incorrect handover decisions, QoS degradation or dropped connections

Capabilities and constraints:

- Attackers **cannot iteratively query** the model
- **No access** to prediction API and confidence scores
- Iterative algorithms are not feasible
- **Outside BTS** attacker is limited to protocol-compliant inputs, and constrained input manipulation, cannot perform black-box optimization attacks (**highly restricted**)
- **Inside BTS** attacker can compute adversarial inputs offline and inject them into data collection pipeline, relies on transferability property, but must ensure that malicious samples are selected for inference
- Inputs are tabular and **structured,** some features cannot be freely modified, and are **constrained** by physical/protocol limits





https://unica-mlsec.github.io/mlsec/slides/03-Evasion-Attacks.pdf

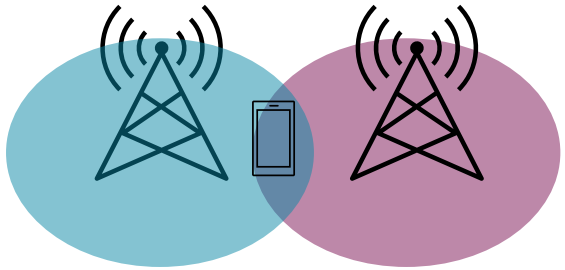
Model Evasion in ML-RAN II

Keep in mind: ML models in RAN are **lightweight** and have lower intrinsic robustness to adversarial perturbations, but inputs are constrained with no query access (**harder to exploit**)

Defenses:

- **System level:** Restrict and validate input resources (UE, IoT, logs), monitor data pipelines
- **Detection-based:** Consistency check on radio, signal measurements, anomaly detection
- **Preprocessing:** Compression, denoising, filtering
- **Adversarial training:** Regularization, smooth optimization, robust policy learning under certain assumptions, ensembles
- **Certified robustness: must be redefined**, distance metrics vs. discrete features, multiple cells, but can be operationally useful

Handover model is certified to keep the same decision if RSRP variation ≤ 2 dB and CQI variation ≤ 1 level



Handover model is certified to keep the same decision if RSRP variation ≤ 2 dB and CQI variation ≤ 1 level

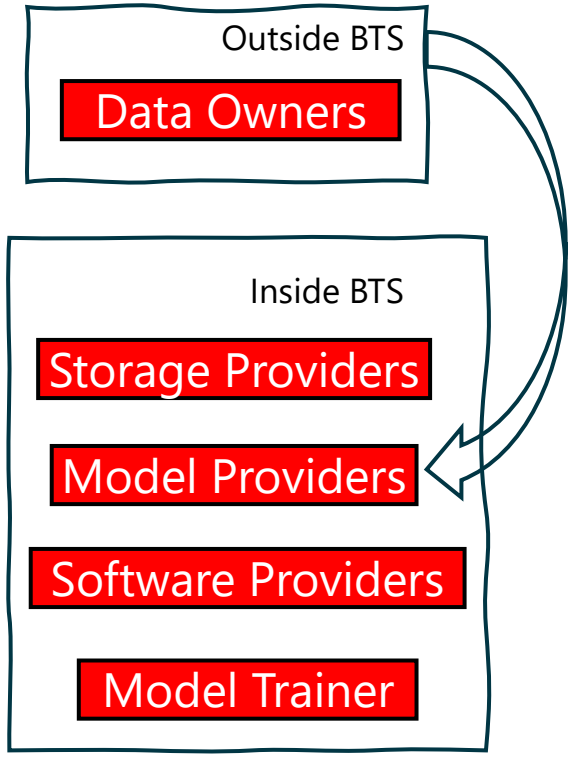
Data Poisoning in ML-RAN I

Attacker profile:

- Outside BTS (e.g., compromised UE, manipulated IoT/NIDD data, corrupted logs or config files) and inside BTS (direct influence on data engine and pipelines)
- **Example goal:** model performance degradation, unstable optimization decisions, bias specific network state, incorrect target cells in handover decisions

Capabilities and constraints:

- Small fraction of the data can be poisoned to remain stealthy
- **Outside BTS** attacker has no access to model, outputs or training process, but can provide poisoned or backdoored data samples with limited capabilities
- **Inside BTS** attacker can inject malicious samples or backdoors to data collection pipeline or modify trained model parameters and config files
- Some features cannot be freely modified, and are **constrained** by physical/protocol limits



Data Poisoning in ML-RAN II

Keep in mind: Outside BTS is an **untrusted zone** and attack success depends on **persistence in data pipelines**

Defenses:

- **System level:** Control and validate data sources (UE, IoT, logs), secure and audit data collection pipelines, dataset versioning and rollback
- **Detection-based:** Outlier detection on radio measurements, anomaly detection over time/cells
- **Preprocessing:** Data sanitization, filtering unreliable or inconsistent inputs, small ensembles to filter decisions
- **Robust training:** Regularization, fine-pruning, model cleansing for backdoors
- **Certified defenses:** There is **no clear epsilon bound** in untrusted zone

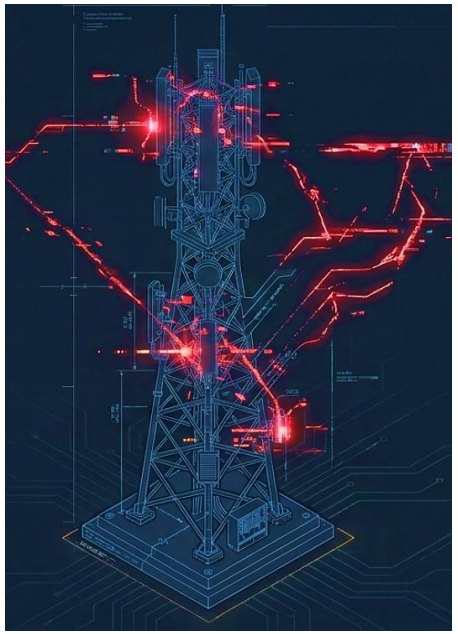


Image created by NotebookLM

Conclusion

- **Theoretical ML Threats:** Assumes unconstrained adversaries, API access, free manipulation of inputs, (unlimited or generous) querying capabilities
- **RAN Application Reality:** Strict protocol rules, physical base station isolation, severe timing and latency limits, zero external API exposure (BTS as closed-loop environment)

- Traditional ML threat models assume **generous adversary capabilities**, while real-world applications operate under **strict physical constraints**
- Attackers and defenders **can be restricted** by physical access, protocol boundaries, operational limits, and input constraints
- Threat modelling for ML-RAN requires evaluating **the full system context**

https://www.ericsson.com/en/blog/2024/8/5g-advanced-handover-triggered-mobility