Finanziato dall'Unione europea
NextGenerationEU

Ministero dell'Università e della Ricerca

Italiadomani
PIANO NAZIONALE DI RIPRESA E RESILIENZA

FAIR Future Artificial Intelligence Research

AI-based approach for classifying the anomalies in the split computing-based system

G. Esposito, E. Magliano, N. Scarano, T. Ahmed Eltaras, J.D. Guerrero Balaguera, L. Mannella, J.E. Rodriguez Condia, A. Ruospo, S. Di Carlo, M. Levorato, A. Savino, M. Sonza Reorda

09/07/2025

Politecnico di Torino

1859

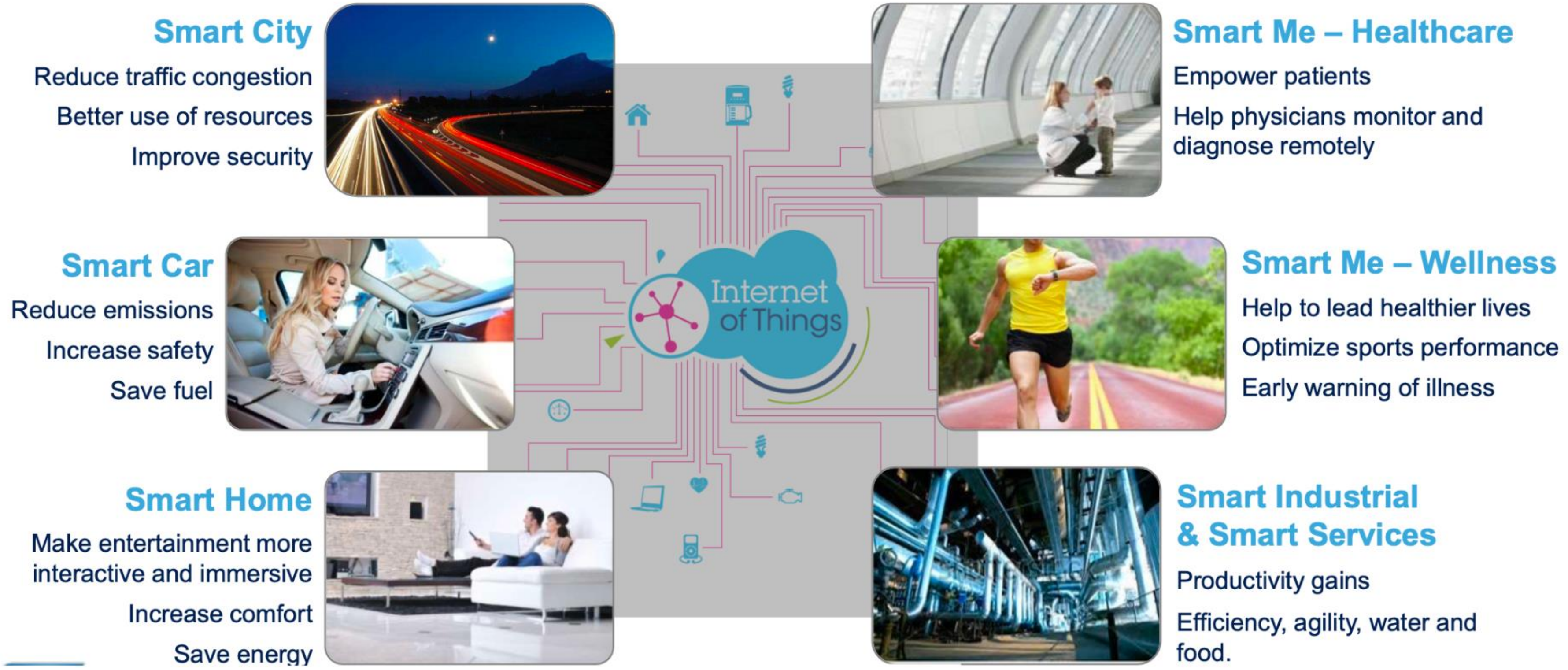
Outline

- Introduction,
- System threat analysis,
- Proposed solution,
- Experimental setup,
- Experimental results,
- Conclusion and future works.

Outline

- Introduction,
- System threat analysis,
- Proposed solution,
- Experimental setup,
- Experimental results,
- Conclusion and future works.

Introduction



Smart City

Reduce traffic congestion
Better use of resources
Improve security

Smart Car

Reduce emissions
Increase safety
Save fuel

Smart Home

Make entertainment more interactive and immersive
Increase comfort
Save energy

Smart Me – Healthcare

Empower patients
Help physicians monitor and diagnose remotely

Smart Me – Wellness

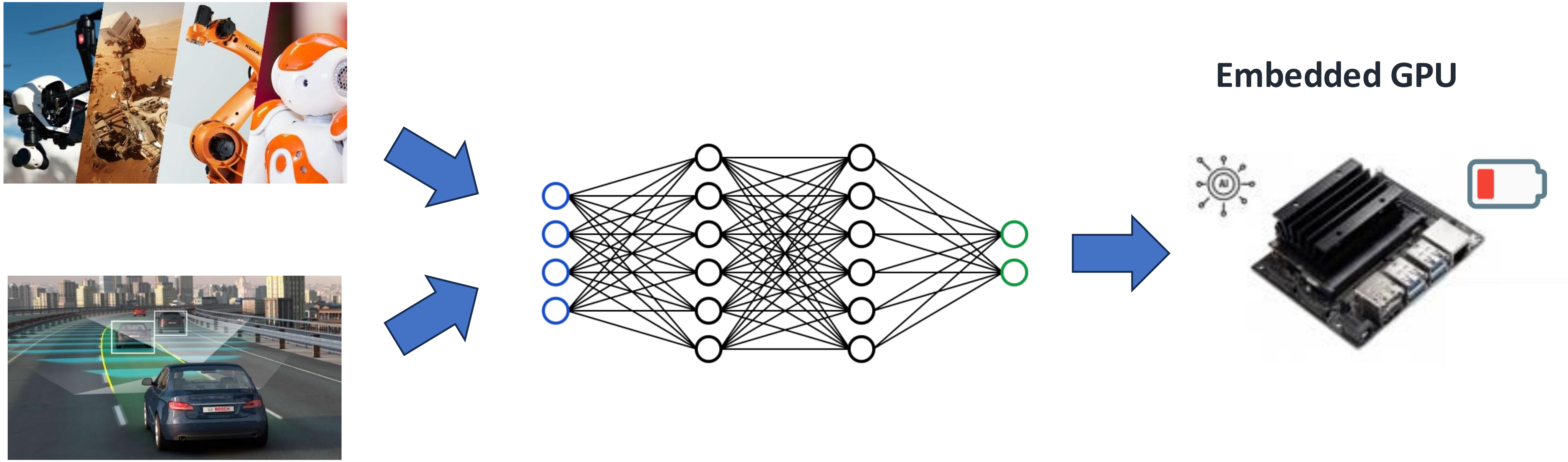
Help to lead healthier lives
Optimize sports performance
Early warning of illness

Smart Industrial & Smart Services

Productivity gains
Efficiency, agility, water and food.

AI is nowadays playing a crucial role in our everyday life.

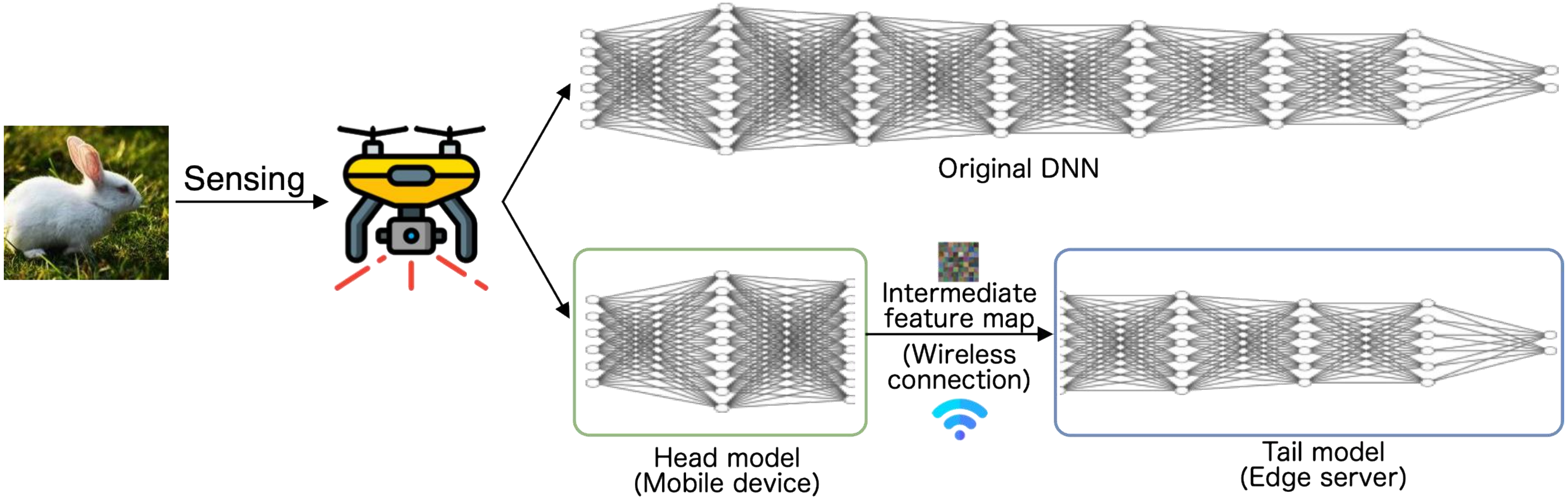
Introduction



However, the complexity of such algorithms often limits the feasibility of their deployement on resource constrained devices, such as embdedded GPUs.

Introduction

Split computing is an NN design paradigm that aims to **optimize** the **power consumption** required from NN inference.



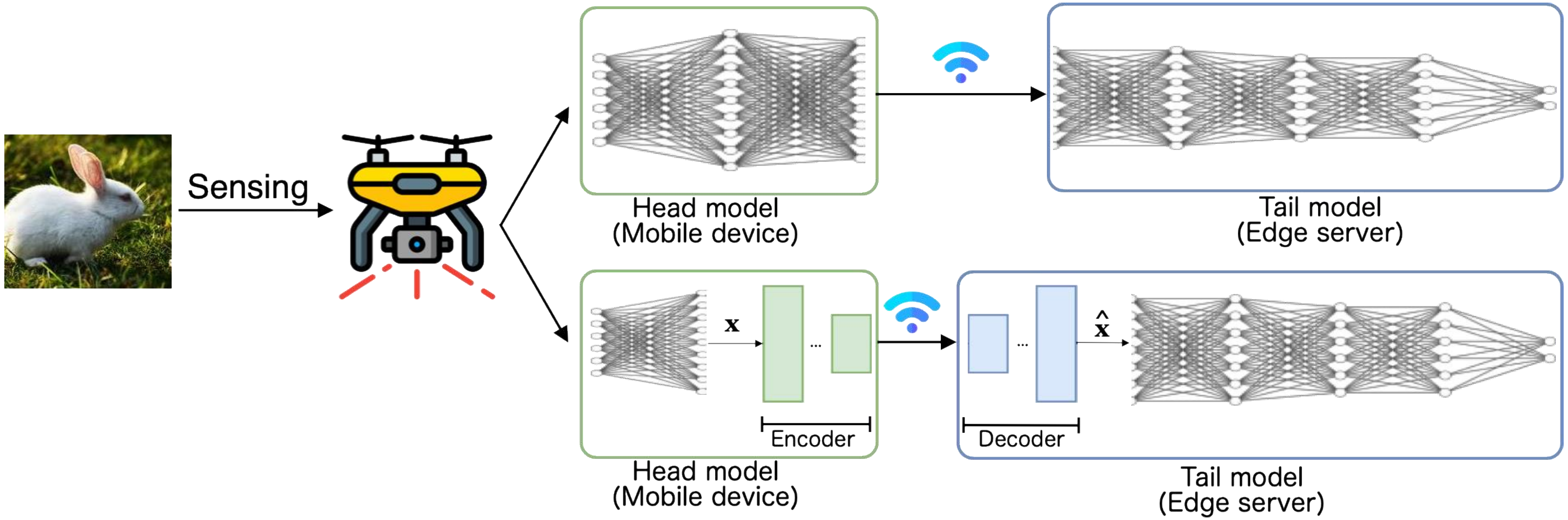
Finanziato dall'Unione europea
NextGenerationEU

Ministero dell'Università e della Ricerca

Italiadomani
PIANO NAZIONALE DI RIPRESA E RESILIENZA

FAIR Future Artificial Intelligence Research

Introduction

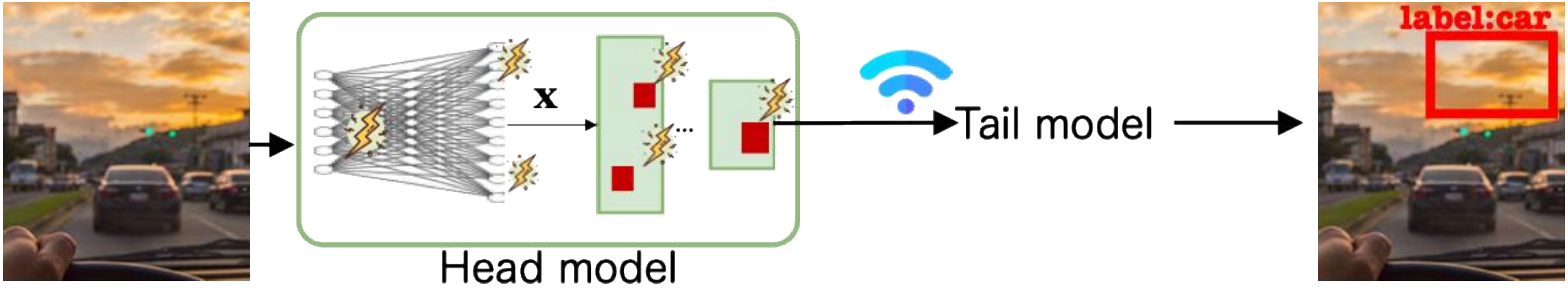


- The bandwidth of the wireless connection poses **limitations** to the transmitted **Feature Map size (FM)**. Thus, an **encoder-decoder architecture** is used to compress this FM.
- However, deploying Split Computing Neural Networks in real safety-critical systems introduces significant challenges concerning **reliability** and **security.**

Outline

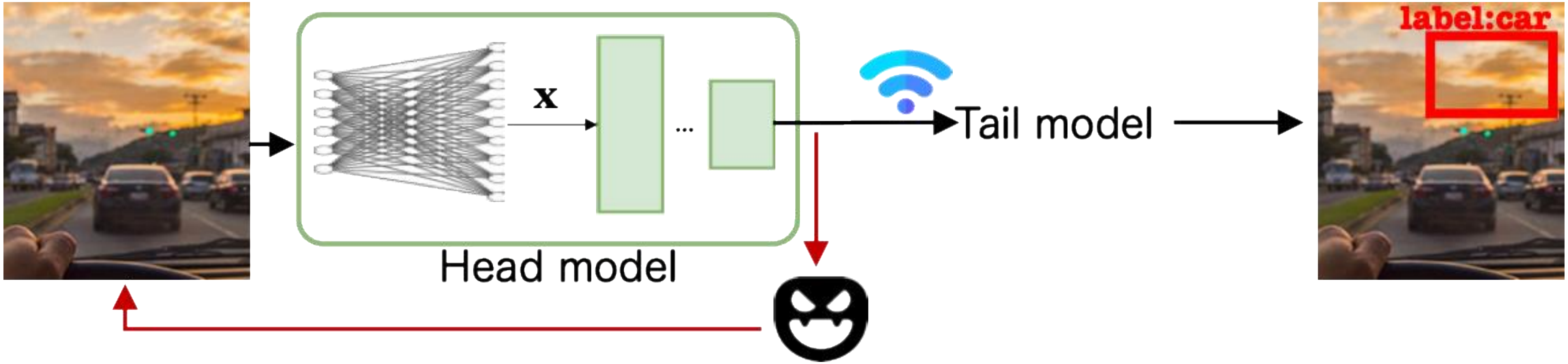
- Introduction,
- System threat analysis
- Proposed solution,
- Experimental setup,
- Experimental results,
- Conclusion and future works.

System threat analysis – reliability issues



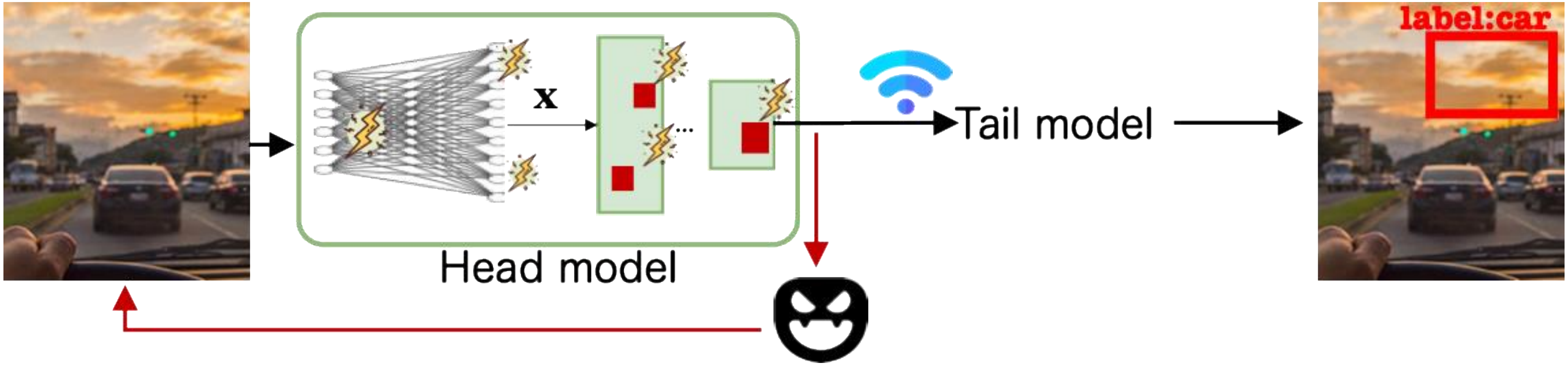
On one side, hardware faults—whether due to aging or external influences like radiation—can trigger critical failures, potentially resulting in catastrophic outcomes, such as car accidents.

System threat analysis – security issues



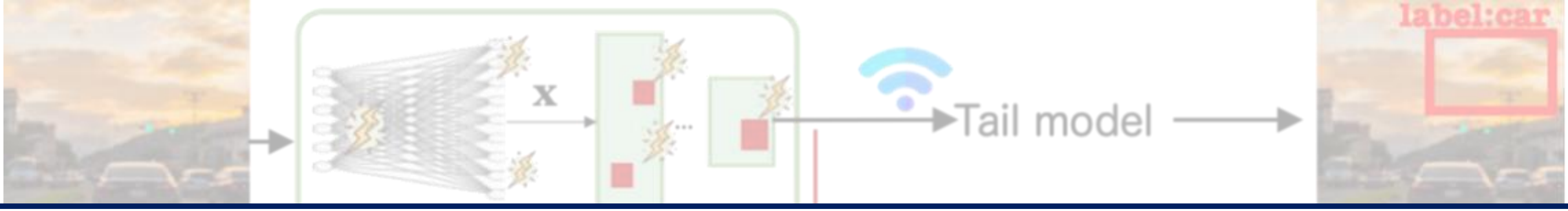
On the other hand, malicious attacks can tamper with internal data during Head–Tail communication, disrupting tasks like object detection and lane keeping—jeopardizing vehicle safety.

System threat analysis



While the **impact** on the Split Computing Neural Network may be similar, the diverse nature of the **threats** (i.e., reliability or security issues) necessitates **tailored countermeasures**.

System threat analysis



Thus, distinguishing between hardware faults and security attacks is essential to enable targeted countermeasures tailored to each specific threats.

While the **impact** on the Split Computing Neural Network may be similar, the diverse nature of the **threats** (i.e., reliability or security issues) necessitates **tailored countermeasures**.

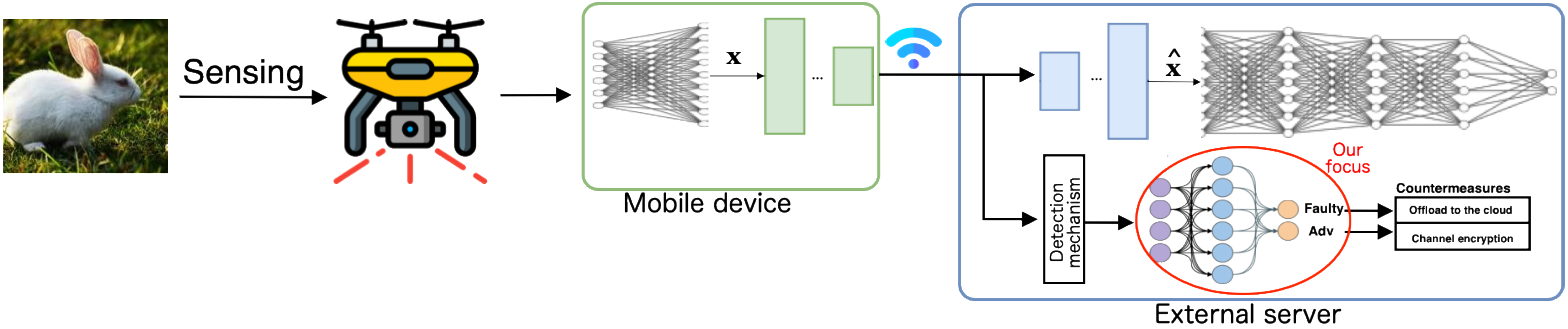
Outline

- Introduction,
- System threat analysis,
- **Proposed solution,**
- Experimental setup,
- Experimental results,
- Conclusion and future works.

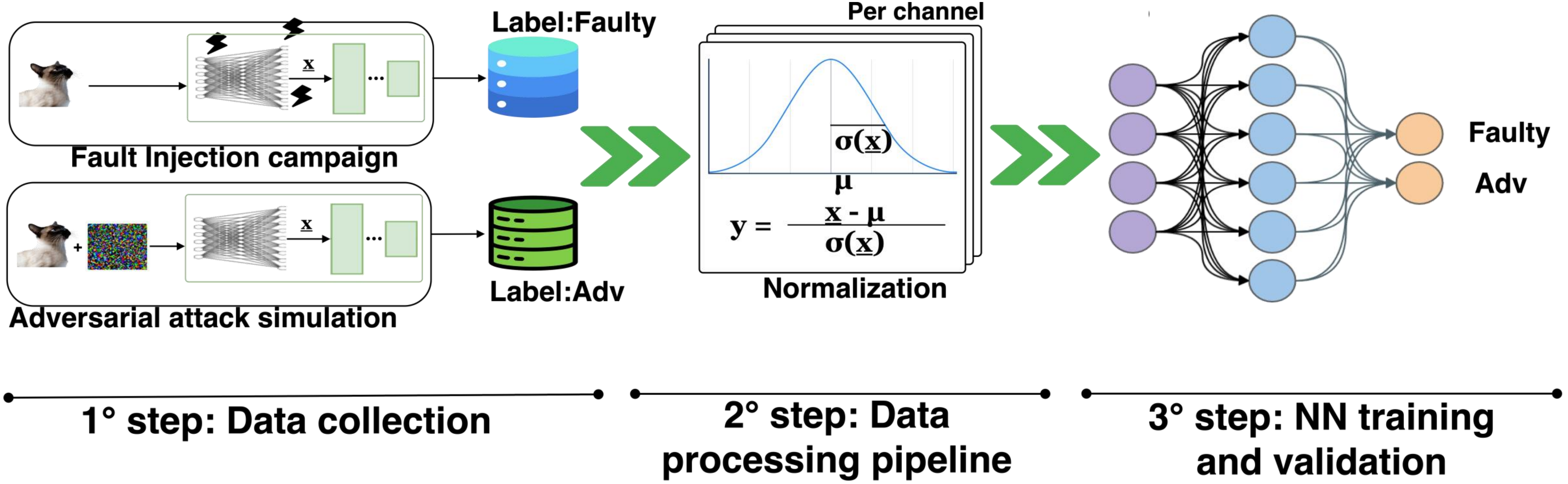
Proposed solution

Thus, including a mechanism that can distinguish between hardware malfunctions and external attacks is crucial to apply proper countermeasures:

- **Reliability** issue countermeasure: the computations can be offloat to the cloud.
- **Security** issue countermeasure: the used channel can be encrypted with a more secure key.



Proposed solution



$$y = \frac{\underline{x} - \mu}{\sigma(\underline{x})}$$

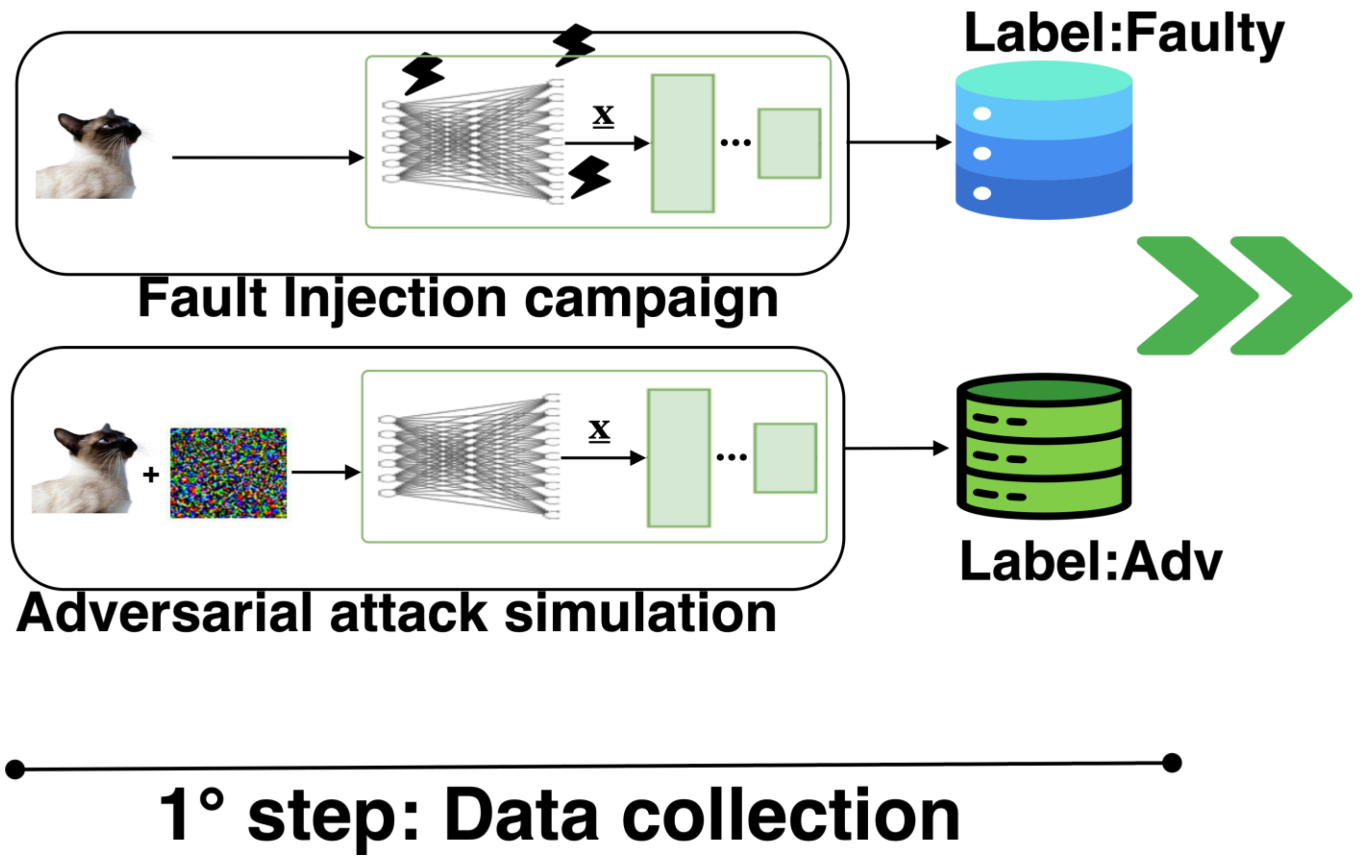
1° step: Data collection

2° step: Data processing pipeline

3° step: NN training and validation

Proposed solution

- Reliability fault model:
  - The Faulty dataset is populated with Fault Injection campaigns that corrupt the neurons' output.
  - The Corruption pattern resort to a hardware-aware error model proposed in [1].
- Security threat model
  - Gray-box model where the attacker can access the Head's output.
  - Based on the computed gradient, the attacker can perturb the Head input.



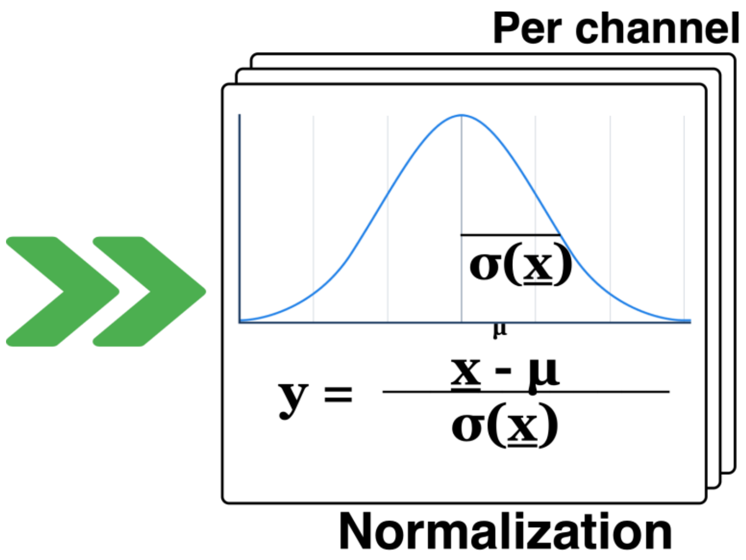
Proposed solution

- Data processing pipeline:

1. To prevent overfitting due to the similar impact of different corruptions on feature maps, we applied a **stratified undersampling** strategy.

2. We implemented **per-channel normalization**, which helps distribute the corruption patterns more evenly across the feature maps. This makes them more distinguishable for the classifier.

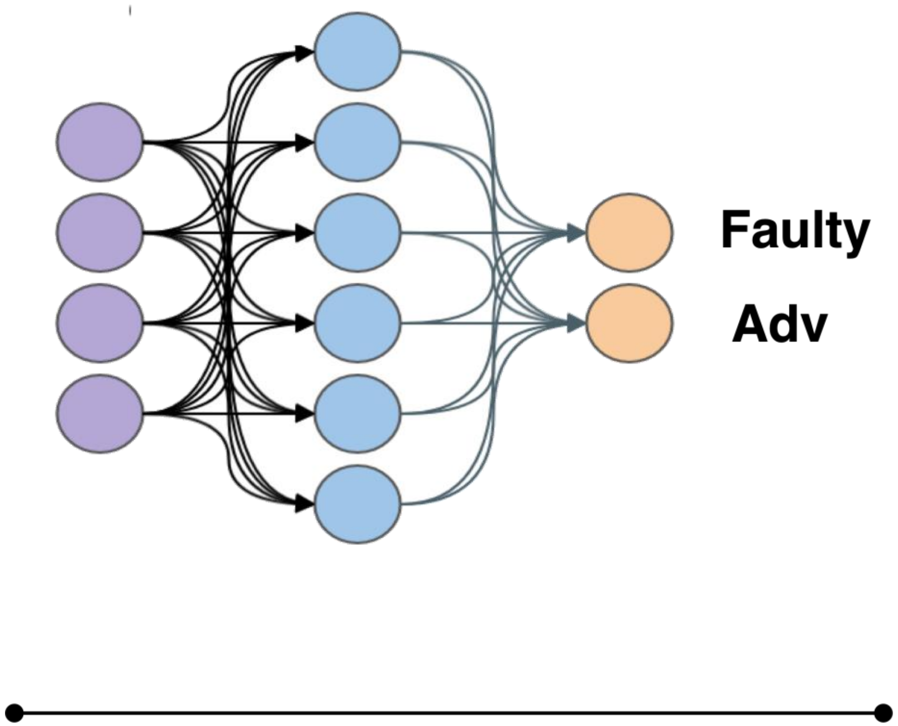
3. Following machine learning best practices, the final dataset—comprising both **reliability-induced** and **security-induced** corruptions—was split into **80% training**, **10% validation**, and **10% testing** subsets.



2° step: Data processing pipeline

Proposed solution

- NN training and validation:
  - The classifier is a Multi Layer perceptron composed of 3 linear layers.
  - During the training process, we employed the Adam optimizer for accurate training convergence to the solution as much close as possible to the optimal one
  - We employed a Cross Entropy Loss to train the Neural Network for the Binary Classification task.



3° step: NN training and validation

Outline

- Introduction,
- System threat analysis,
- Proposed solution,
- Experimental setup,
- Experimental results,
- Conclusion and future works.

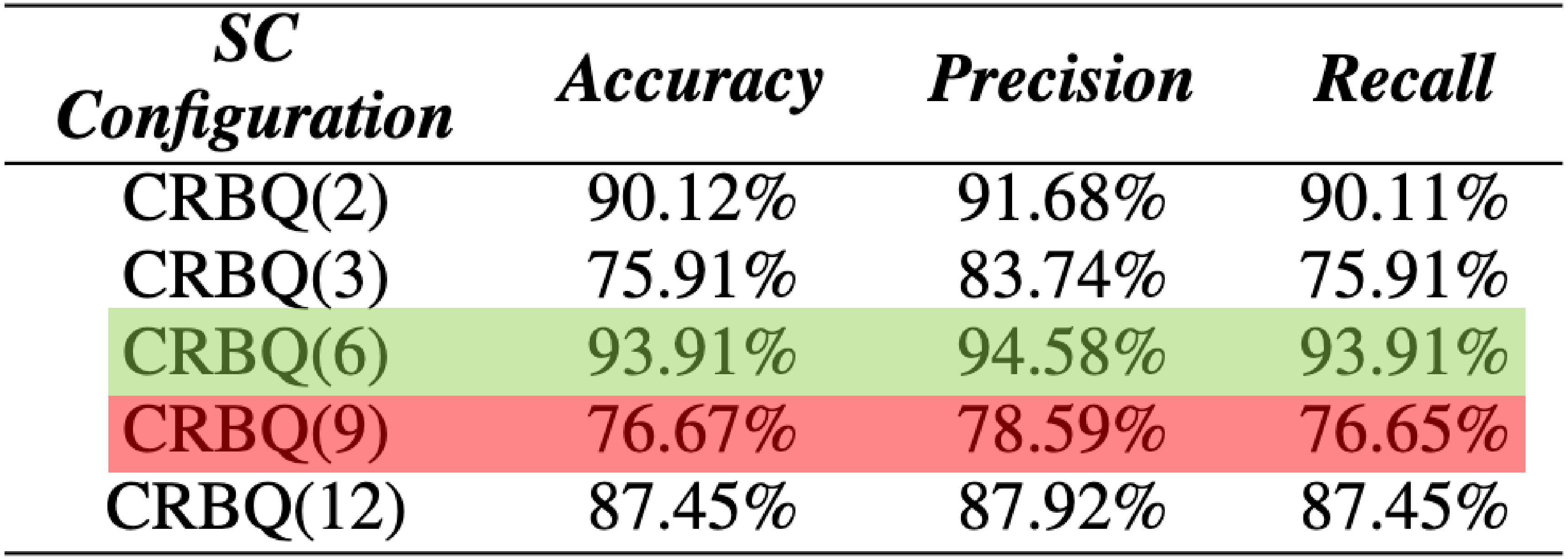
Experimental setup

- **Simulation Environment**: SC-DNN architectures and pretrained models from [1] were used to simulate SC systems locally.
- **Model Configuration**: Five CRBQ-configured ResNet-50 models tested on ILSVRC 2012 dataset.
- **Corruption Campaigns**: A hardware-aware error model was used to inject faults and a gradient-based algorithm was used to simulate adversarial attacks on SC Head using 50 representative ImageNet samples.
- **Fault Injection Details**: 1,600 experiments per FI campaign, Bit Error Rates ∈ [0.004, 0.1], and exponent bit location ∈ [19, 31].
- **Adversarial Attacks**: 15,000 perturbed samples generated via 300 attack iterations per image.
- **Undersampling Strategy**: Balanced class sizes and removed redundant samples to improve robustness and slow convergence.
- **Dataset Split**: 4,183 training / 1,791 validation / 1,792 test samples.

Outline

- Introduction,
- System threat analysis,
- Proposed solution,
- Experimental setup,
- Experimental results,
- Conclusion and future works.

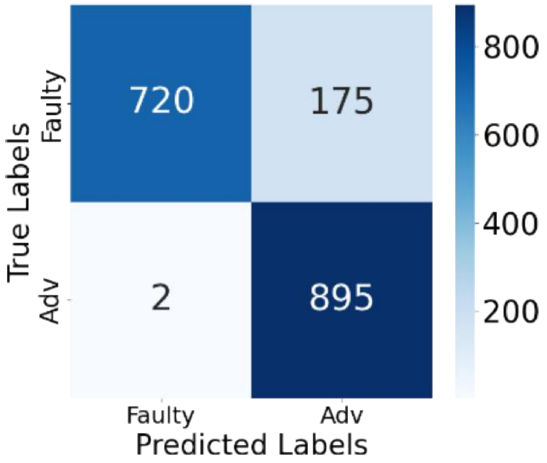
Results



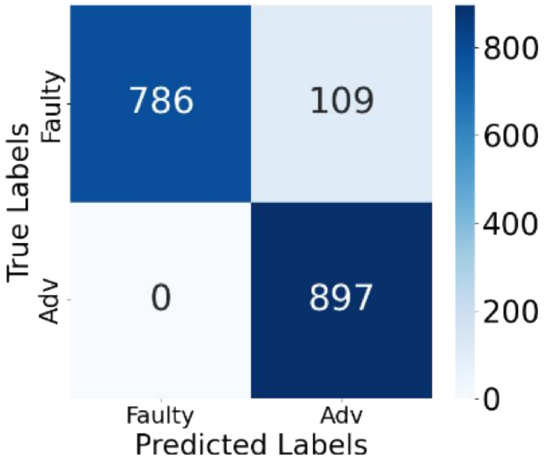
SC Configuration	*Accuracy*	*Precision*	*Recall*
CRBQ(2)	90.12%	91.68%	90.11%
CRBQ(3)	75.91%	83.74%	75.91%
CRBQ(6)	93.91%	94.58%	93.91%
CRBQ(9)	76.67%	78.59%	76.65%
CRBQ(12)	87.45%	87.92%	87.45%

Split configuration CRBQ(6) enables the best classifier performance being capable of effectively distinguishing between adversarial attacks and hardware faults 93.91% of the times.

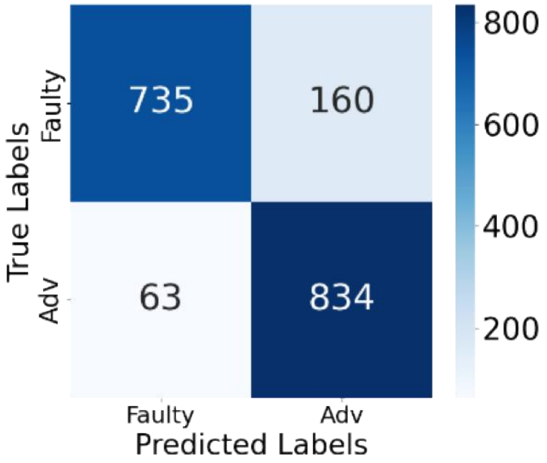
Results



(a) CRBQ (2)



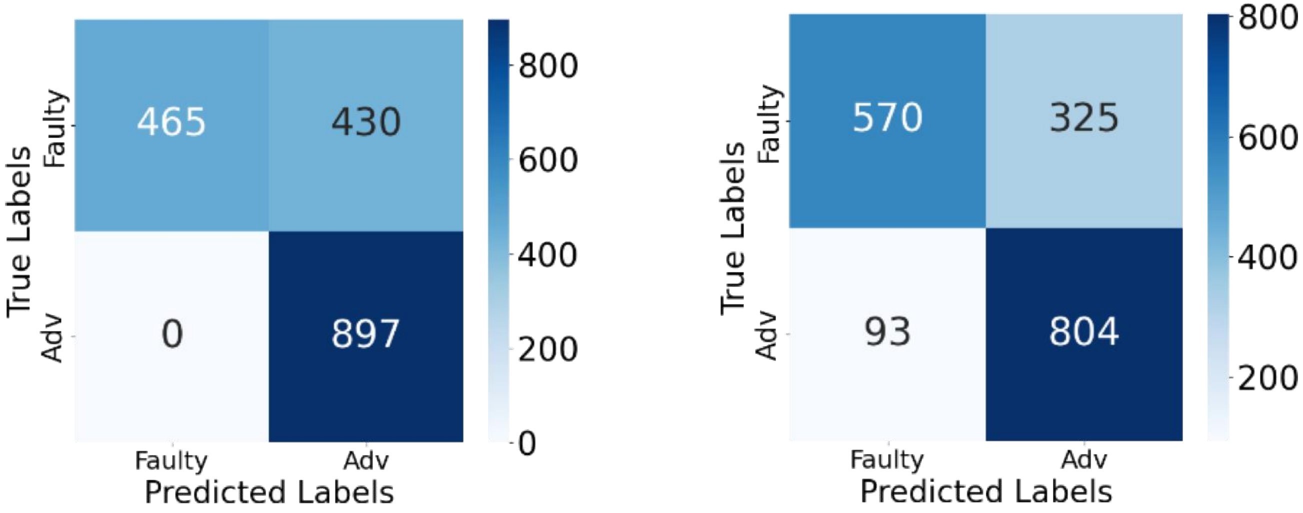
(c) CRBQ (6)



(e) CRBQ (12)

Split configurations CRBQ(2) CRBQ(6) CRBQ(12) shows the highest balance between True Positive Rate and True Negative Rate identifying a tendancy of classifying «Faulty» samples as «Adv» samples.

Results



(b) CRBQ (3)

(d) CRBQ (9)

The missclassification of «faulty» samples as «Adv» samples is more visible in CRBQ(3) and CRBQ(5).

Outline

- Introduction,
- System threat analysis,
- Proposed solution,
- Experimental setup,
- Experimental results,
- Conclusion and future works.

Conclusions and future works

- This work presents, for the first time, a solution for distinguishing between adversarial attacks and hardware-induced corruptions in SC systems.
- The Neural Network-powered classifier achieved up to 93.91% accuracy.
- Our study reveals that CRBQ(6) is the Split Computing configuration that better enable the implementation of the system capable of distinguishing between Hardware Faults and Adversarial Attacks.
- As future works we plan to extend the selection of Split Computing configurations to different compression setups.
- We also plan to explore alternative classification and anomaly detection strategies prioritizing their explainability of the models.

Finanziato dall'Unione europea
NextGenerationEU

Ministero dell'Università e della Ricerca

Italiadomani
PIANO NAZIONALE DI RIPRESA E RESILIENZA

Future Artificial Intelligence Research

THANKS!
Question?

G. Esposito, E. Magliano, N. Scarano, T. Ahmed Eltaras, J.D. Guerrero Balaguera, L. Mannella, J.E. Rodriguez Condia, A. Ruospo, S. Di Carlo, M. Levorato, A. Savino, M. Sonza Reorda

Politecnico di Torino

09/07/2025