Commonwealth Cyber Initiative

O-RAN Forensics: Data-Driven Reconstruction of Conflicts with Graph Neural Networks

Joao F. Santos
Research Assistant Professor
Virginia Tech

Network Automation & Self-Organization

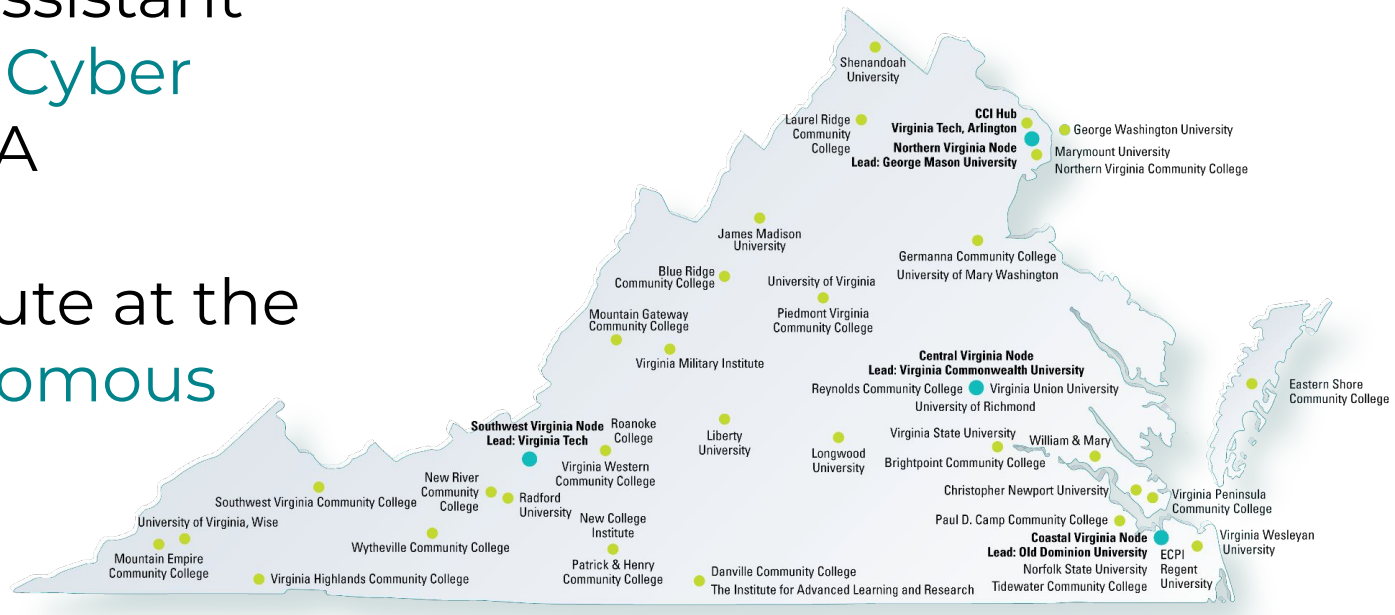
BOWW | Berlin, DE | Sep 9th, 2025

Who am I

I am Joao F. Santos, I am research assistant professor with the Commonwealth Cyber Initiative at **Virginia Tech** in the USA

We are a distributed research institute at the intersection of cybersecurity, autonomous systems, and network intelligence

My background is in software-defined wireless communications, focusing in experimental research on wireless networks, from software-defined radios to programmable network architectures



CCI VIRGINIA NETWORK

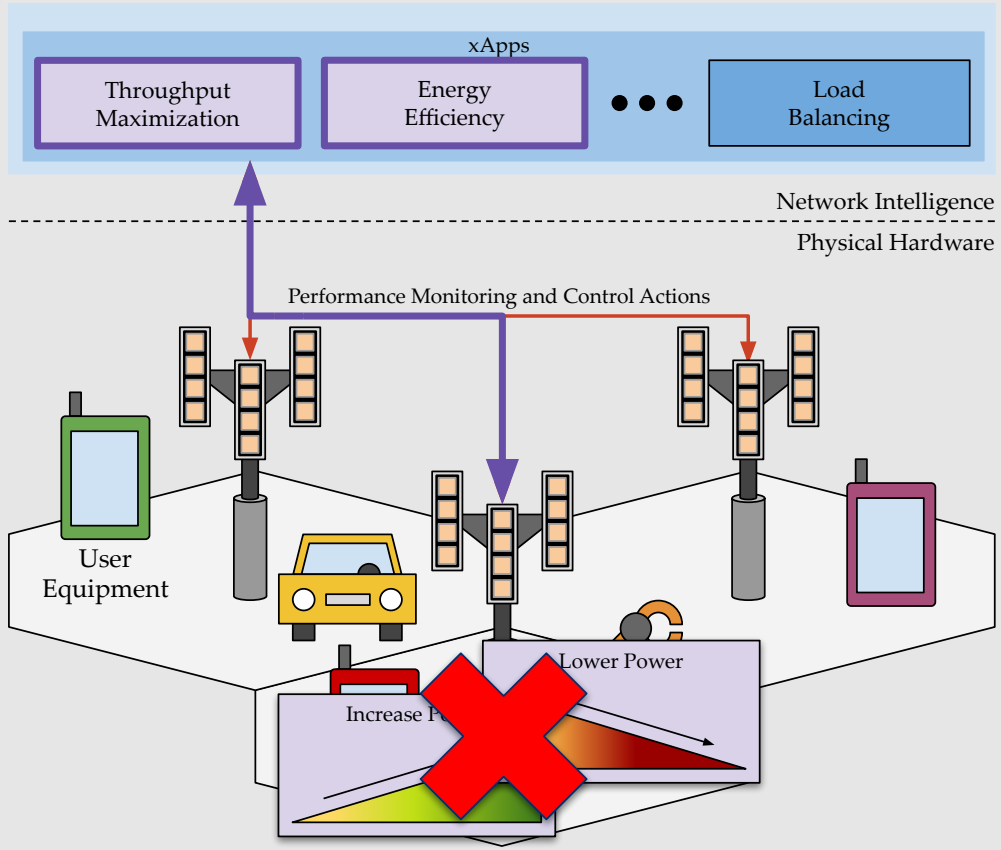
47 Higher Education Institutions
300+ faculty members

Open Networks… And Prone to Conflicts

Mobile networks are increasingly managed by independent control functions, e.g., xApps, rApps and dAps in the context of O-RAN, or more broadly, a collection of AI agents

However, multiple control functions may attempt to modify the same RAN control parameters to achieve distinct outcomes, creating potential conflicts

These conflicts are internal vulnerabilities that can degrade performance[1], cause instability, or even disrupt network operation



Examples of potential conflicting interactions between xApps with different objectives

[1] del Prever, Pietro Brach, et al. "Pacifista: Conflict evaluation and management in open ran." *IEEE Transactions on Mobile Computing*, 2025.

Open Problems, Ongoing Research

While the O-RAN Alliance recognizes the need for a Conflict Mitigation service in the Near-RT RIC[2], there are still no standardized methods to achieve mitigation – or even conflict detection

There is a very small but growing academic literature on conflict modeling and detection for xApps in O-RAN

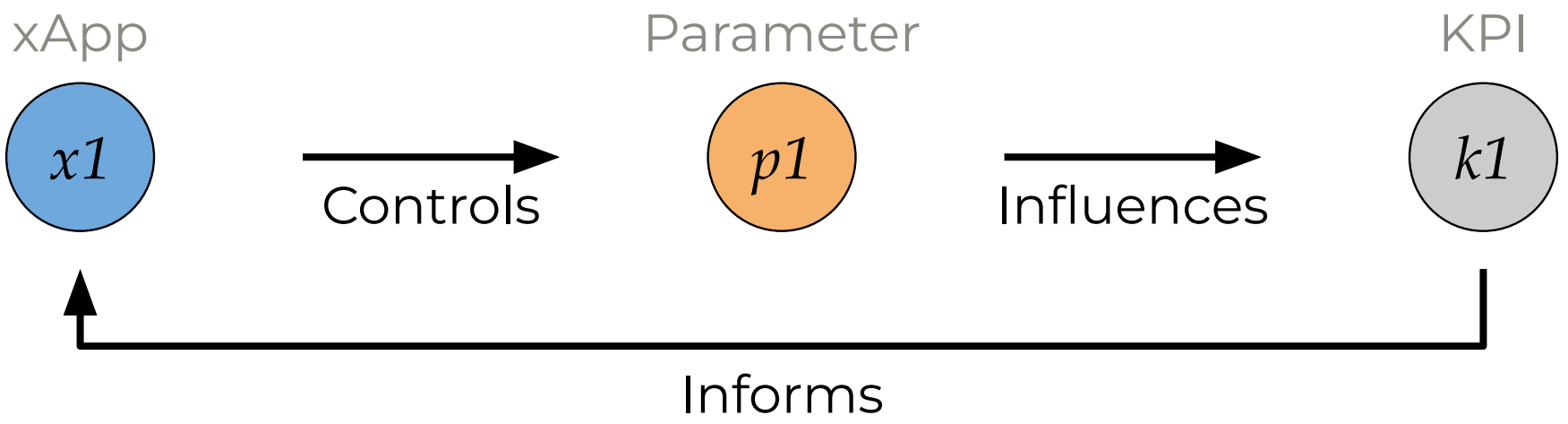
Existing works are effective, but often focus on one type of conflict, rely on limiting assumptions (e.g., all xApps are cooperative DRL agents), or depend on sandbox environments and handcrafted testing suites (by network admins)

However, there is a lack of general, data-driven, and autonomous methods for detecting conflicts between xApps in O-RAN

[2] O-RAN Working Group 3, "Near-RT RIC Architecture," O-RAN Alliance, Tech. Rep., Feb. 2025.

Graph-based Conflict Modeling

Our approach, based on Graph Theory[3] is to model the relationships between RAN control parameters, KPIs, and xApp actions as graphs:



This forms a heterogeneous graph structure that can represent potential conflicts – a conflict graph

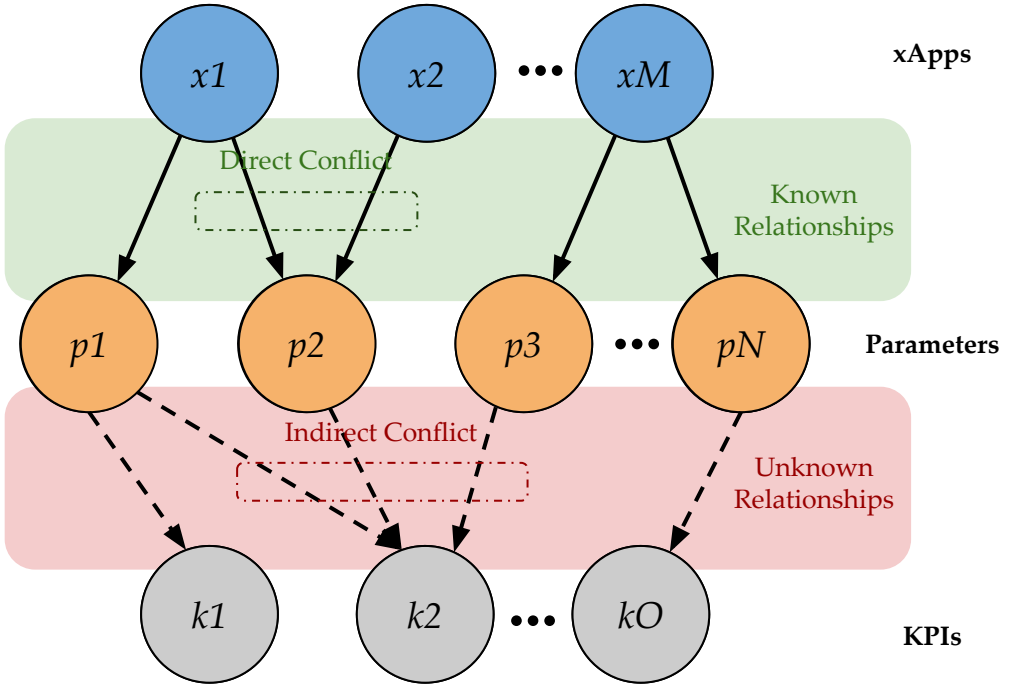
[3] Arshia Zolghadr, Joao F. Santos, Luiz A. DaSilva, and Jacek Kibilda, "Learning and Reconstructing Conflicts in O-RAN: A Graph Neural Network Approach", IEEE Wireless Communications and Networking Conference (WCNC) Workshops, Milan, Italy, 24–27 Mar. 2025.

Conflict Graphs

This general modelling approach captures relationships for any xApp (rule- and AI-based)

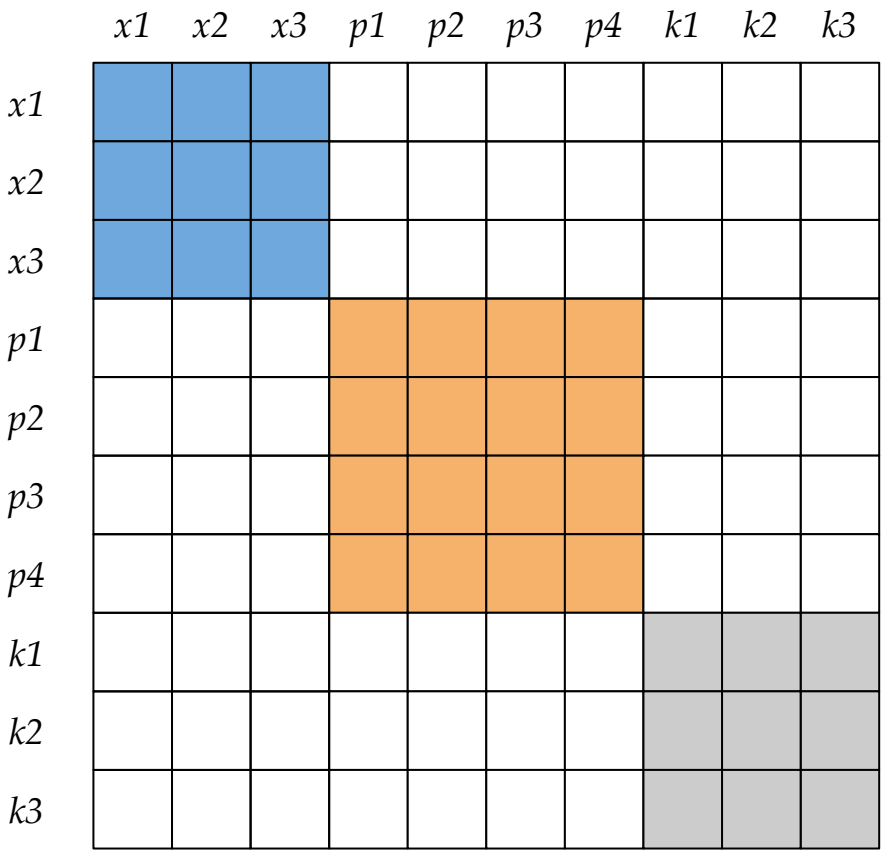
By analyzing the topology of the conflict graph, we can identify all types of conflict (direct, indirect, and implicit), i.e., looking at incident edges or chains of dependencies

The challenge lies in constructing the conflict graph, as some relationships between xApps, control parameters and KPIs are known by design before deploying xApps, but others may not be known priori



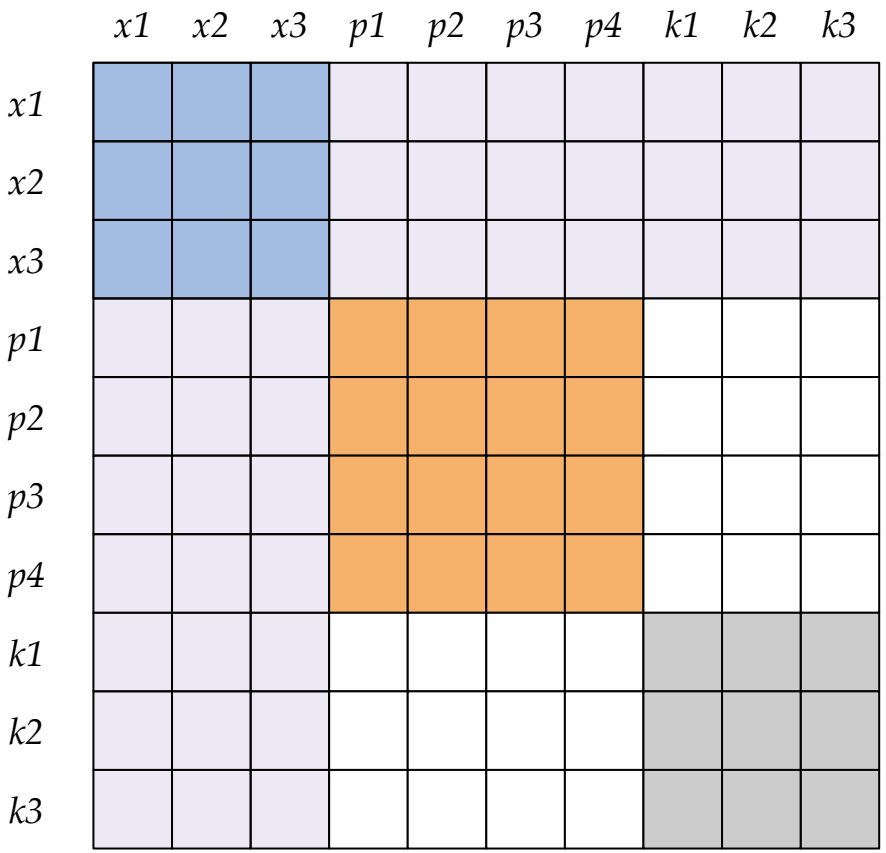
Graph-based representation of the relationships between xApp, control parameters and KPIs

(In)Complete Adjacency Matrix



If we represent the conflict graphs as an adjacency matrix A, we can observe regions of the matrix populated by xApps (blue), control parameters (orange) and KPIs (gray)

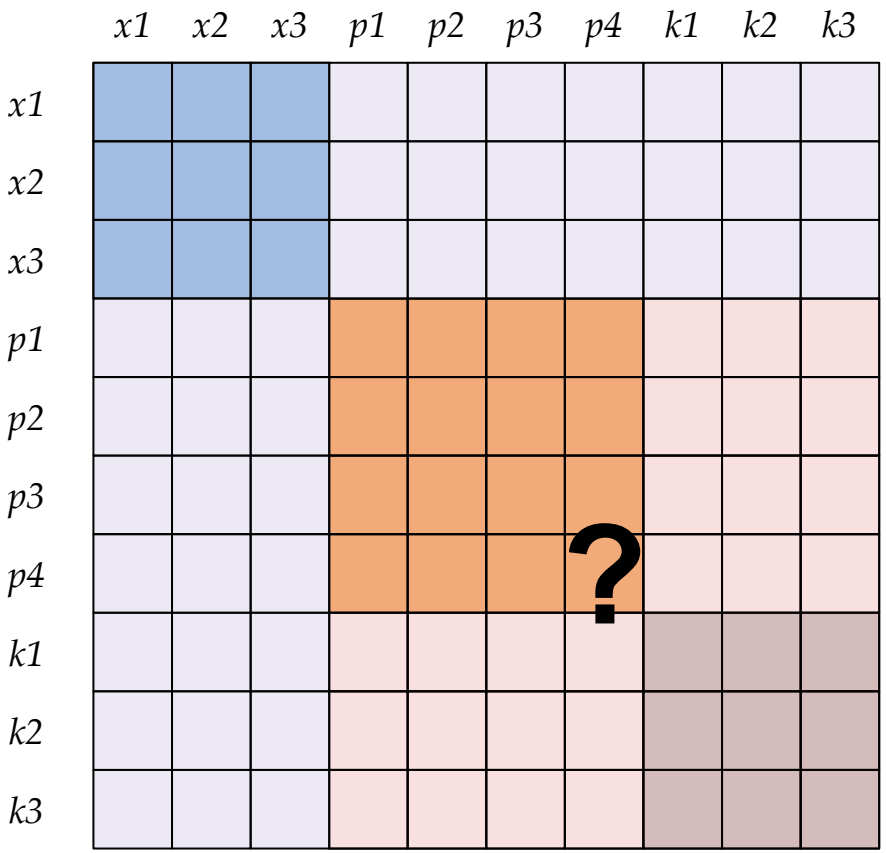
(In)Complete Adjacency Matrix



If we represent the conflict graphs as an adjacency matrix A, we can observe regions of the matrix populated by xApps (blue), control parameters (orange) and KPIs (gray)

There are parts of the matrix that represent the parameters that the xApps control and the KPIs that they consume (purple), which we can obtain a priori as part of the xApp subscription process

(In)Complete Adjacency Matrix



If we represent the conflict graphs as an adjacency matrix A, we can observe regions of the matrix populated by xApps (blue), control parameters (orange) and KPIs (gray)

There are parts of the matrix that represent the parameters that the xApps control and the KPIs that they consume (purple), which we can obtain a priori as part of the xApp subscription process

However, the relationship between control, parameters and KPIs (red) can be scenario-dependent, dynamic, and non-trivial, and in some cases, is not even known

Graph Neural Networks to the Rescue

...meaning that our adjacency matrix is incomplete ⚠️

And without a complete adjacency matrix**, we cannot create a reliable conflict graph** to represent and **detect all conflicts in a radio access networks**

To address this challenge, we proposed[3] a data-driven approach based on Graph Neural Networks (GNNs) for learning the relationships between control parameters and KPIs based on data collect from the RAN

We can use GNNs to predict links and complete the adjacency matrix, allowing us to reconstruct conflict graphs, and finally detect conflicts autonomously

Recap on GNNs

GNNs are a family of neural networks designed to work with graph-structured data. They can be very powerful tools for[4]:

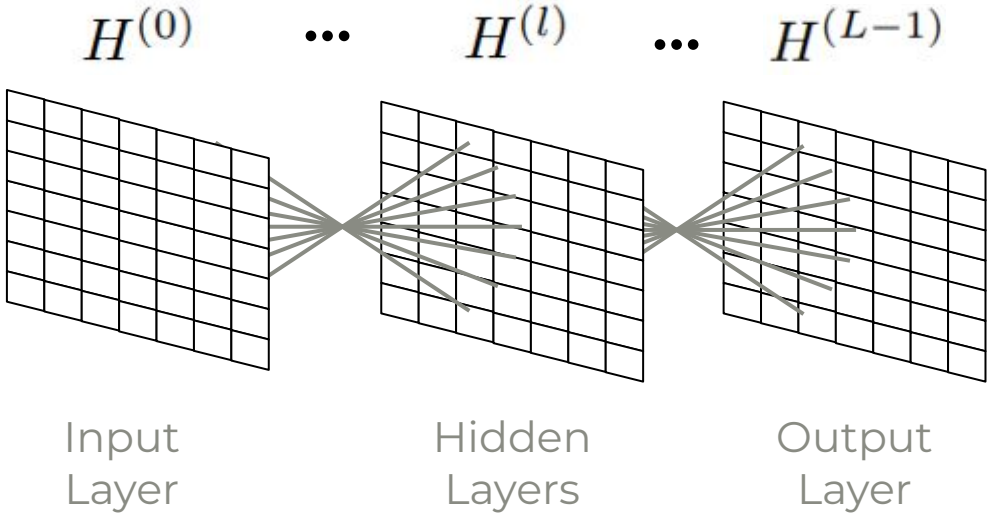
- Graph Classification: Predict properties for entire graphs (e.g., molecule toxicity)
- Node Classification: Predict node labels based on neighbors (e.g., fraud detection)
- **Link Prediction: Predict missing edges** (e.g., recommend friends)

By leveraging GNNs for link prediction, we can identify hidden relationships and predict links between control parameters and KPIs, allowing us to complete the adjacency matrix and reconstruct conflict graphs, and finally detect conflicts autonomously

[4] Liu, Xingyu, Juan Chen, and Quan Wen. "A Survey on Graph Classification and Link Prediction Based on GNN." arXiv preprint arXiv:2307.00865 (2023).

Graph-Learning with GNNs

For a conflict graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with control parameters and KPIs as nodes $\mathcal{V}$ and their values as the input feature matrix $H^{(0)}$



We can use GraphSAGE, an inductive learning framework, to iteratively learn relationships between nodes based on their effect on one another over time

The embeddings $H^{(l)}$ contain compressed information about the nodes ' own features, local its neighborhood structure, and the strength of its correlation with neighbors

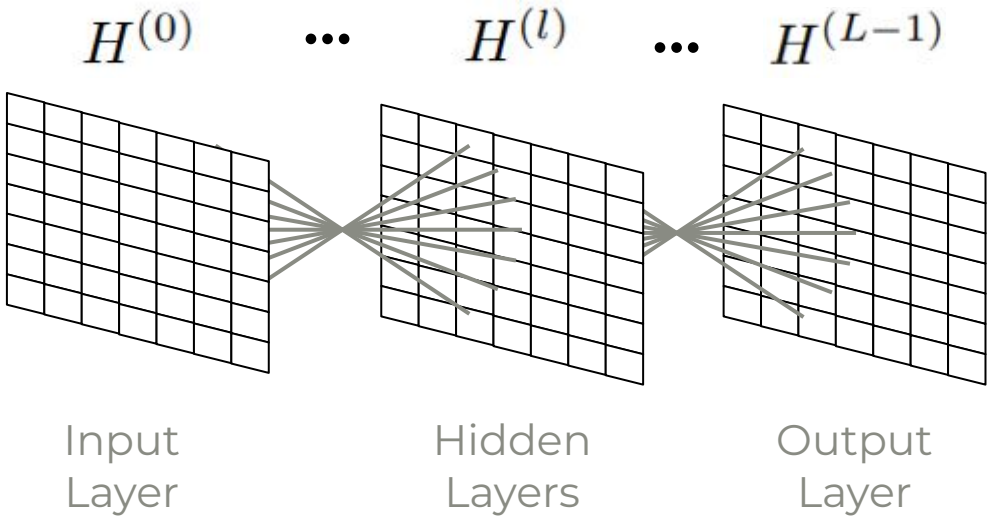
Graph-Learning with GNNs

Each layer of the GNN model updates node embeddings by aggregating information from neighbors using a trainable weight matrix W[5]:

$$H^{(\ell)} = \sigma\Big(\text{CONCAT}\big(H^{(\ell-1)}, \text{AGGREGATE}(A, H^{(\ell-1)})\big)W^{(\ell)}\Big)$$

And $H^{(L-1)}$ represents the output layer, with final embeddings of nodes in latent space

Strongly correlated nodes have similar embeddings, while weakly correlated nodes are pushed apart



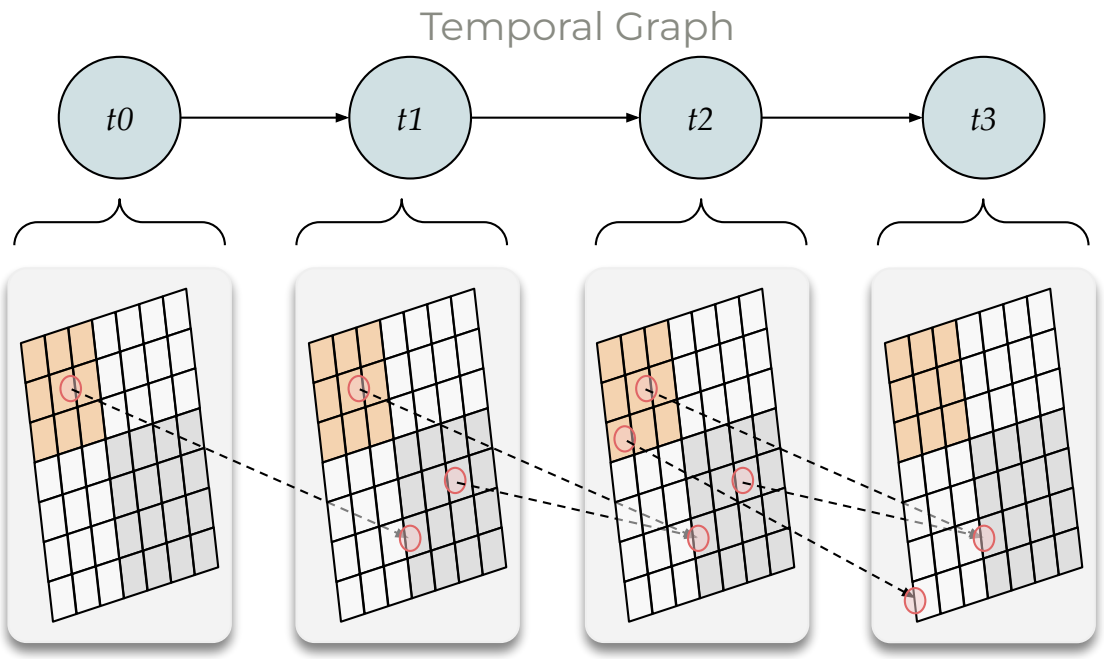
We then apply a dot product decoder between different nodes $s_{ij} = (h_i^{(L)})^\top h_j^{(L)}$ as a learned correlation measure between nodes

[5] W. Hamilton et al., "Inductive Representation Learning on Large Graphs," Advances in Neural Information Processing Systems (NeurIPS), vol. 30, 2017.

Dataset of Network States

We feed our model with a dataset a temporal graph $\mathcal{G}_T = (\mathcal{V}_T, \mathcal{E}_T)$, containing snapshots of the state of the network, with values of the control parameters and KPIs over time

Through training, our GNN identifies which control parameters and KPIs are related to one another, uncovering potential dependencies and interactions between variables that could lead to conflicts



Snapshots of the states of parameters and KPIs at tn

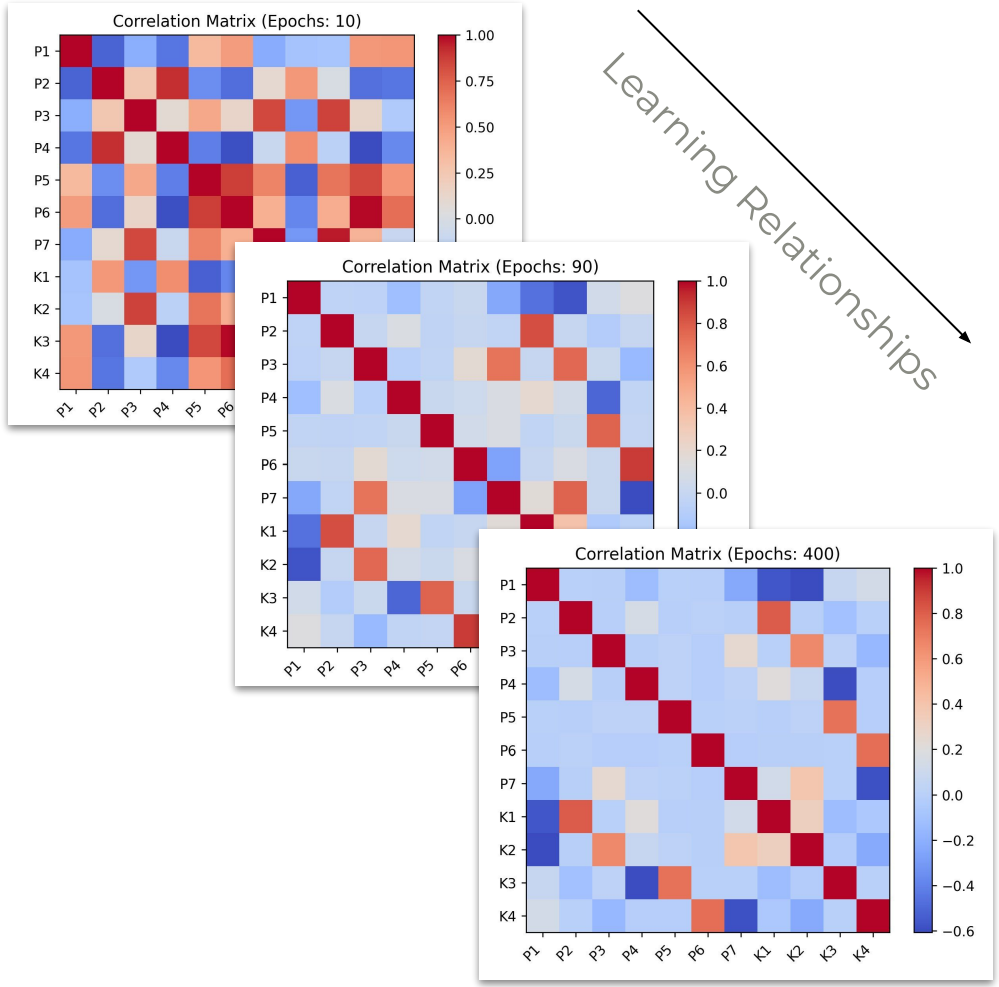
Training and Reconstruction

From a random start, our GNN learns embeddings for each node that capture correlations, then using these embeddings to predict likely links

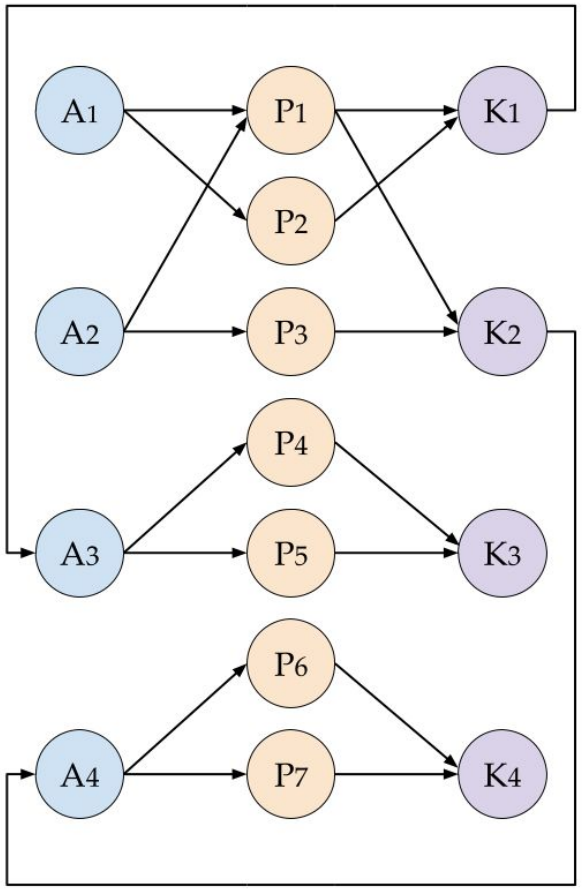
Our model outputs a correlation matrix showing how different nodes are correlated to one another

We can then apply a heuristic threshold **T** to binarize the correlations and obtain a reconstructed adjacency matrix

$$A_{ij} = \begin{cases} 1 & \text{if } |R_{ij}| \geq \tau,\ i \neq j \\ 0 & \text{otherwise} \end{cases}$$



Conflict Model



Structure of the Conflict Model

To validate our approach, we leveraged a conflict model available in the conflict management literature[5], to generate a sample dataset with Gaussian distributed samples

We designed a GNN model with 11 features (4 KPIs and 7 parameters) representing the structure of the conflict model, and trained it to learn the correlations of the sample dataset

[5] A. Banerjee et al., "Toward Control and Coordination in Cognitive Autonomous Networks," IEEE TNSM, vol. 19, no. 1, pp. 49–60, 2021.

Reconstruction Performance

To evaluate our accuracy, we use the F1 Score, the harmonic mean of precision and recall

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}},$$

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}},$$

$$\text{F1 Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}.$$

We evaluated[3] our model's accuracy for reconstructing conflict graphs under different dataset sizes, training epochs, and threshold values, and validated our data-driven approach

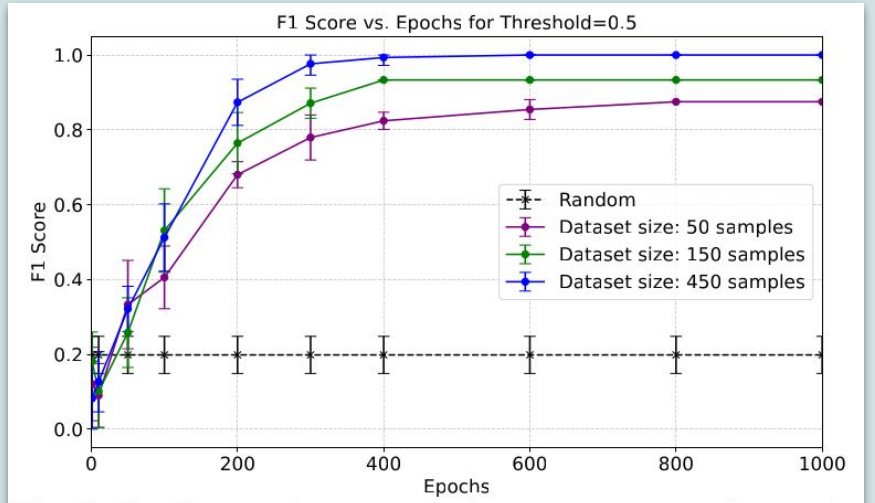


Fig. 5: Conflict graph reconstruction accuracy according to the number of epochs and dataset size for a fixed threshold of 0.5.

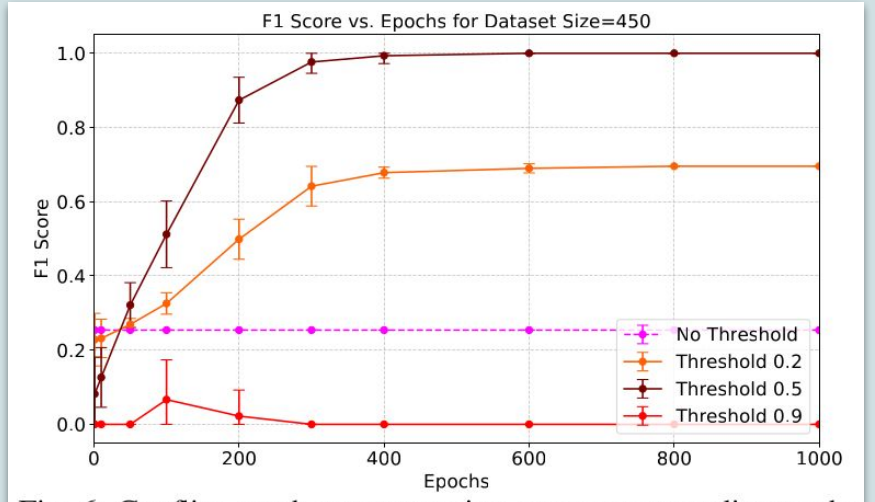


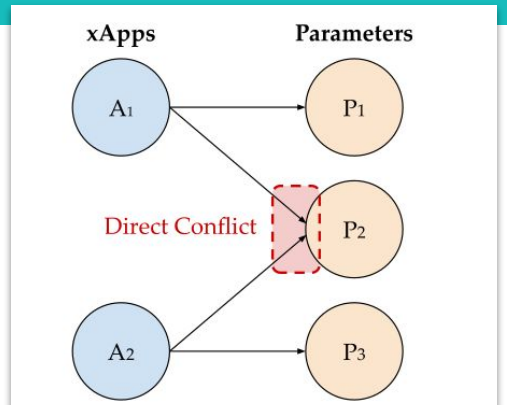
Fig. 6: Conflict graph reconstruction accuracy according to the epochs and thresholds for a fixed dataset size of 450 samples.

[3] Zolghadr et al., "Learning and Reconstructing Conflicts in O-RAN: A Graph Neural Network Approach", IEEE WCNC Workshops, Milan, Italy, 24–27 Mar. 2025.

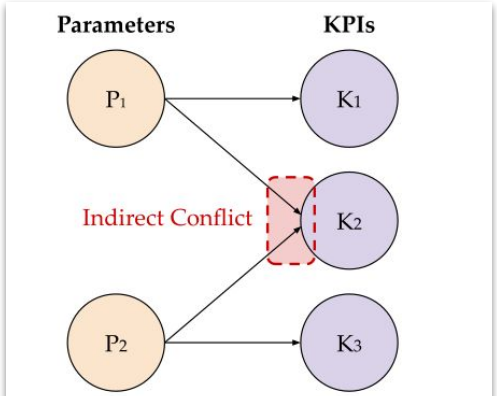
Graph-based Conflict Definition

We also proposed graph-based definitions of the conflicts considered by the O-RAN Alliance, and utilized graph labeling to identify different types of conflicts:

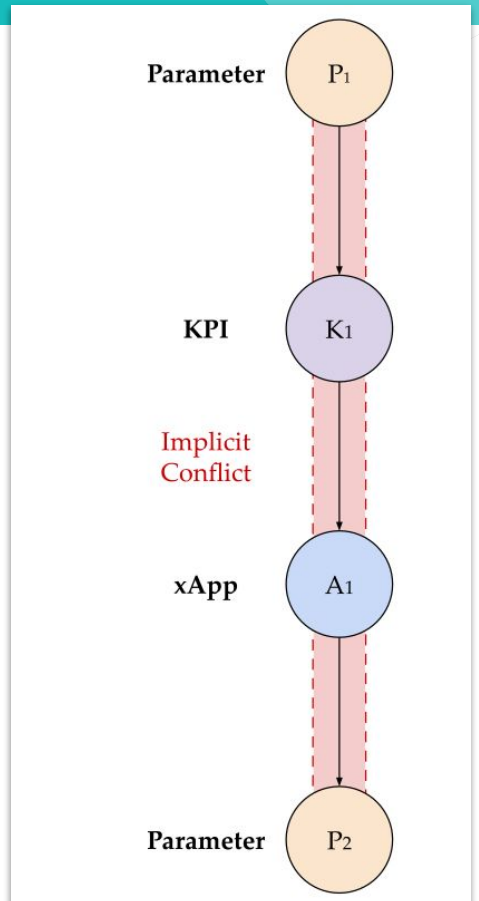
- **Direct**: multiple xApps controlling the same parameters
- **Indirect:** multiple parameters affecting the same KPIs
- **Implicit:** complex chains of dependencies between parameters



(a) Direct conflict between xApps modifying the same control parameters.



(b) Indirect conflict between control parameters affecting the same KPIs.



(c) Chains of complex relationships create a logical dependency and cause an implicit conflict between different parameters.

Conflict Detection Performance

We also evaluated[1] the performance of our graph labelling to detect indirect and implicit conflicts
(Direct conflicts are trivial)

We can observe the importance of the threshold for binarization, and the contributions of the dataset size and training times

We can achieve an 100% detection accuracy when the conflict graph is reconstructed with 450 samples and a 0.5 threshold after 600 epochs

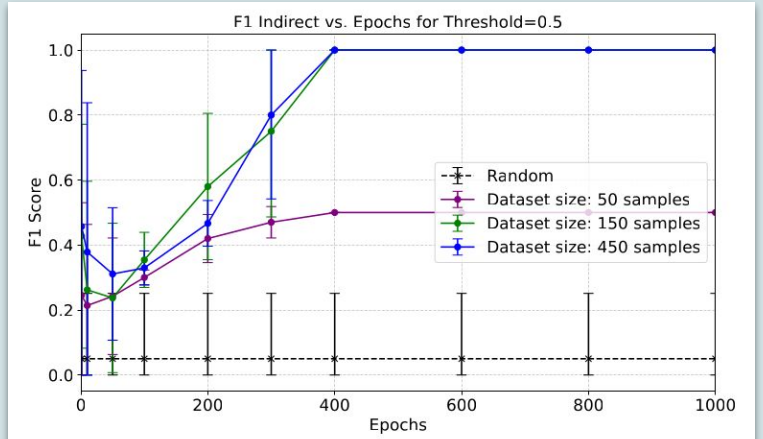


Fig. 7: Indirect conflict labeling accuracy according to the number of epochs and dataset size for a fixed threshold of 0.5.

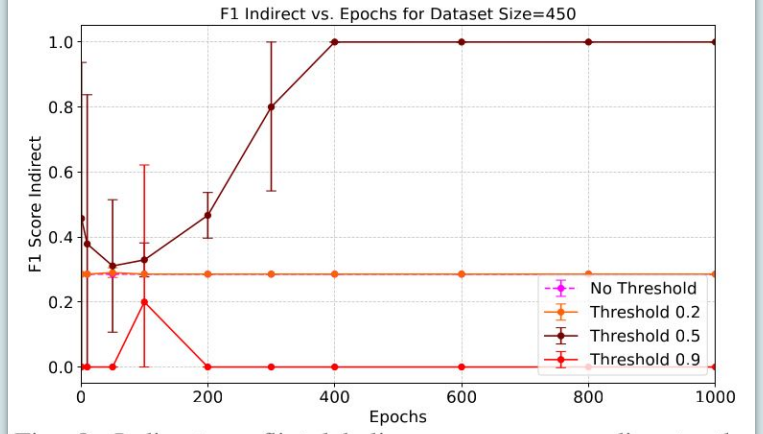


Fig. 8: Indirect conflict labeling accuracy according to the epochs and thresholds for a fixed dataset size of 450 samples.

Indirect Conflicts

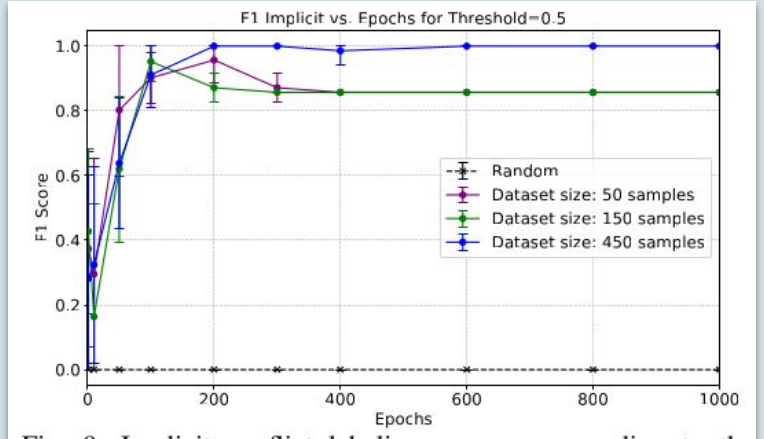


Fig. 9: Implicit conflict labeling accuracy according to the number of epochs and dataset size for a fixed threshold of 0.5.

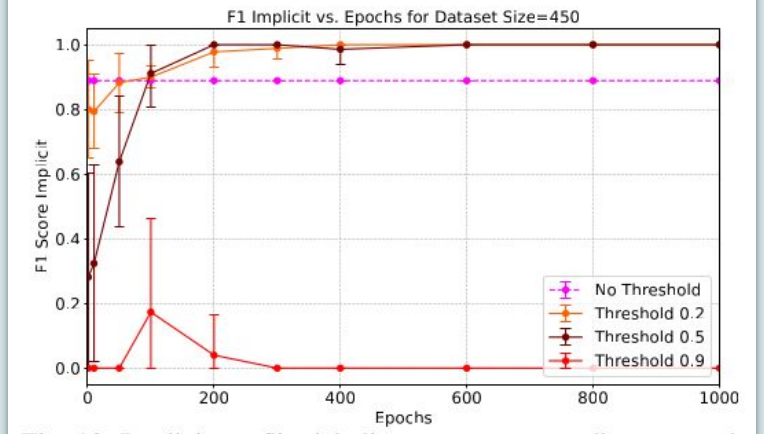


Fig. 10: Implicit conflict labeling accuracy according to epochs and threshold values for a fixed dataset size of 450 samples.

Implicit Conflicts

Current State

- We have proposed a general, data-driven approach for learning relationships and identifying correlations between control parameters and KPIs based from collected data from the RAN
- We can accurately reconstruct the heterogeneous conflict graphs and autonomously detect different types of conflicts using graph-based conflict definitions and graph-labeling

Acknowledgements:
NSF US-Ireland R&D, Award No. 2421362
NSF IUCRC WISPER, Award No. 2412872
DoE INL LDRD, Award No. 25A1090-129FP
6G SNS JU 6G-XCEL, Award No. 101139194

Next Steps

Short Term Future

- Investigating alternative GNN architectures, e.g, Graph Convolutional Networks (GCNs) and Graph Attention Networks (GATs)
- Exploring data-driven methods for autonomously selecting the binarization threshold
  Early results for both, working on a publication ✅

Mid/Long Term Future

- Experimental validation using real data from an actual O-RAN deployment
  Facing issues due to limitations of current real radio stacks ⚠️
  Interested in pivoting to pursue simulation/digital twins

Commonwealth
Cyber Initiative

Questions?