Self-Organizing Cellular Radio Access Network with Deep Learning

***Wuyang Zhang*[1], Russell Ford[2], Joonyoung Cho[2], Charlie Jianzhong Zhang[2], Yanyong Zhang[1,3], Dipankar Raychaudhuri[1]**

[1] WINLAB, Rutgers University
[2] Samsung Research America
[3] University of Science and Technology of China

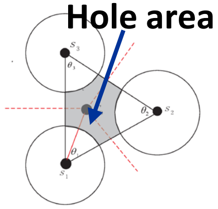
IEEE INFOCOM Workshop 2019

Problem Statement

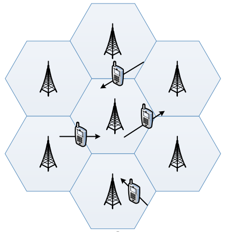
- **RAN Performance Problems Prevalent**
  - *My phone shows 5 signal bars but the connection is so slow!*
  - *Cannot hear your voice!*
  - *This web page is not loading at all!*

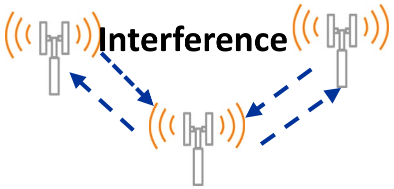
- **Example root causes of RAN performance problems**

Excessive uptilt/downtil (EU/DU)

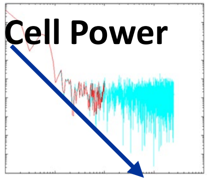


Coverage hole (CH)

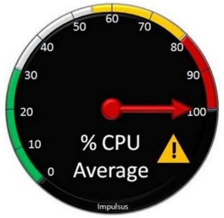

Too late handover (TLHO)



Inter-cell Interference (II)



Excessive Cell Power Reduction (ERP)

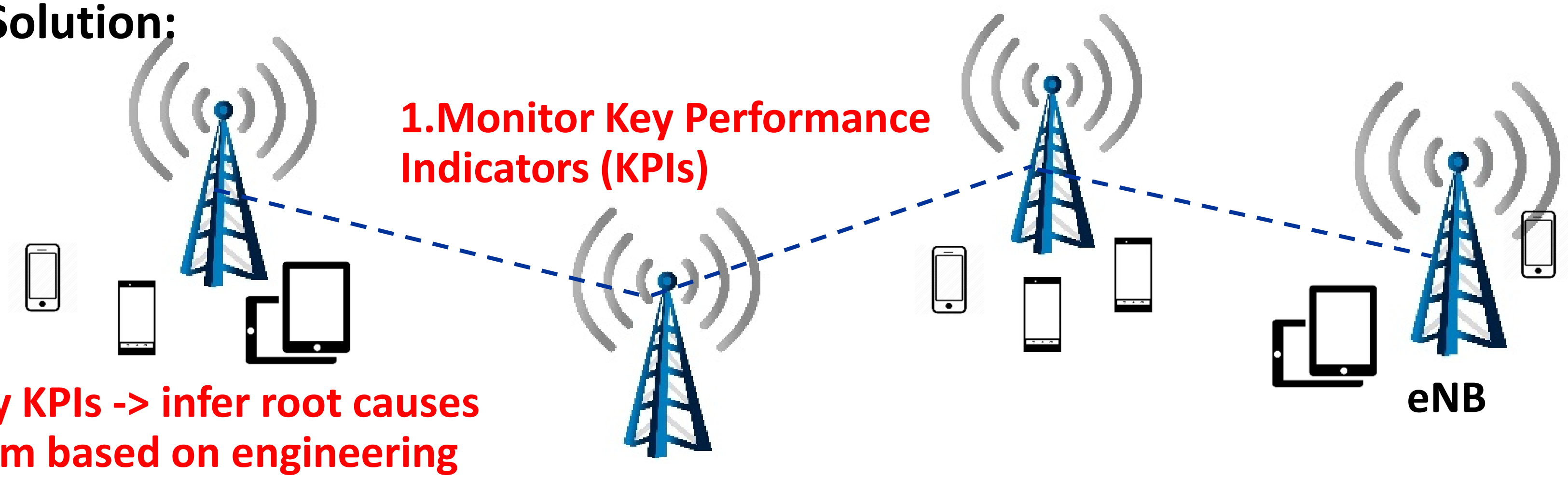


Cell Overload (CO)

- **NOT straightforward to diagnose the root cause!**

Self-Healing Radio Access Network

Existing Solution:



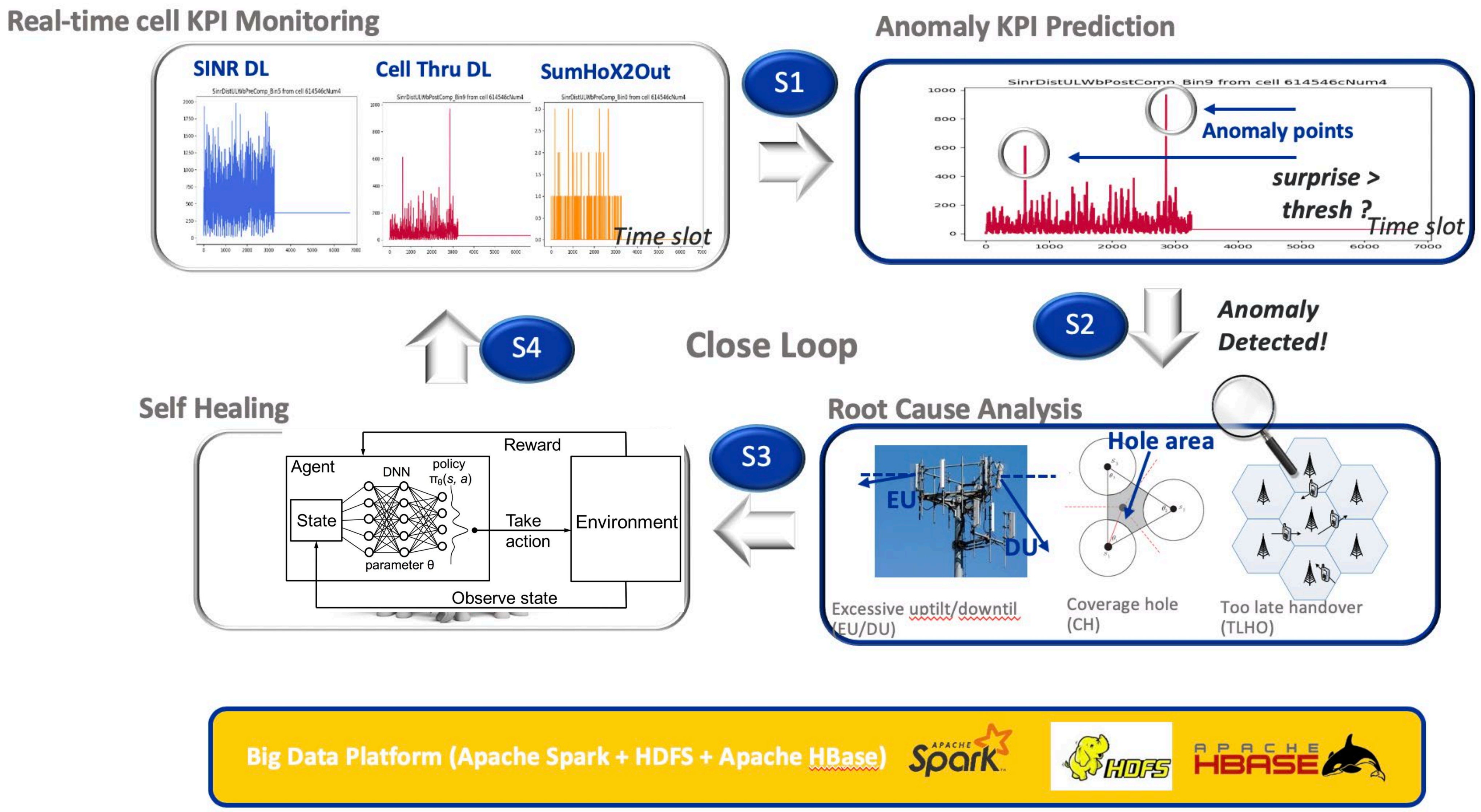
2. Anomaly KPIs -> infer root causes and fix them based on engineering experience （take hours)

Can cellular network operators automate the diagnosis and self-healing of RAN?

System challenges:

- *How to **predict anomaly KPIs** before any faults really appear?*
- *How to figure out **root causes** based on thousands of cell KPIs?*
- *How can the system **self recover** from the faults?*
- *How to deal with **~ TB level*** data of *cell KPIs?*

System Overview



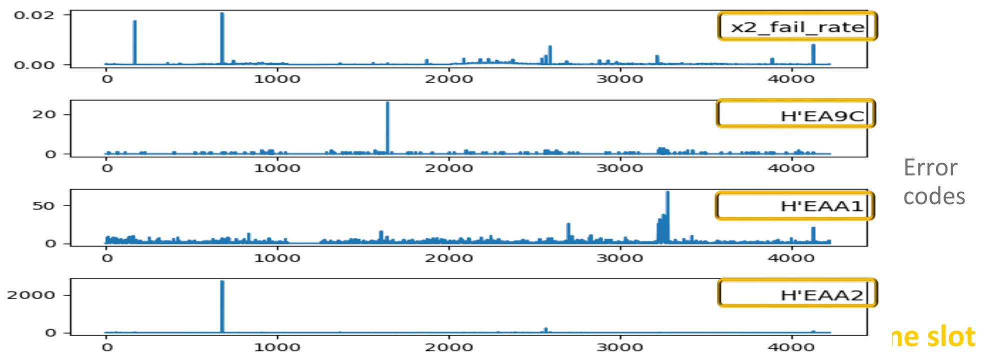
RUTGERS WINLAB | Wireless Information Network Laboratory

Real-world KPI Dataset Overview

real-word data from a top-tier US cellular operator

- **Aggregated Cell Dataset**
  - *KPIs & error code summary: **~100**. e.g., mobile subscriber count, attach count, detach count, handover count; x2_attempt, x2_enb_to, x2_dns_fail, s1_intra_src_attempt, s1_intra_tar_sgw_chg, etc,.*
  - *Overall size: ~335 GB*
  - *Collection date: 2017-06-30 – 2018-03-20*
  - *Collection interval: 1hour*

Example KPIs in time series



- **Non-aggregated Cell dataset**
  - *KPIs summary: **~4k**.*
  - *Overall size: ~ 100 TB*
  - *Collection date: 2018-02-01 – 2018-07-31*
  - *Collection interval: 15 minutes*

- Accessibility
- Retainability
- Integrity
- Availability
- Mobility
- Connection Drop Rate
- Cell Throughput

Partial example KPIs

'InterferencePowerAvg', 'InterferencePowerTot', 'InterferencePowerCnt', 'ThermalNoisePowerAvg', 'ThermalNoisePowerTot', 'ThermalNoisePowerCnt', 'RssiOverPathAvg', 'RssiOverPathTot', 'RssiOverPathCnt', 'RssiPath0Avg'

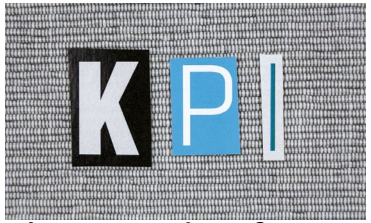
RUTGERS
WINLAB | Wireless Information Network Laboratory

Anomaly Prediction: Objective & System Challenges

Objective: based on the currently/historically-reported cell KPIs, to ***predict the potential anomaly KPIs/events*** in the future

System Challenges:

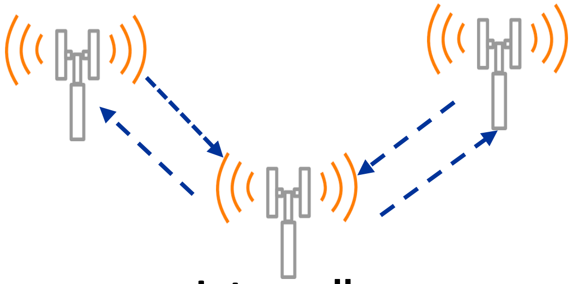
- ***Identify related KPIs:*** Difficult to know in advance which of the **thousands** of KPIs are **relevant and correlated** with the predictive KPIs.
- ***Inter-cell interference***: Some KPIs from neighboring cells may be related, like in the case of high inter-cell interference, but may not trigger an anomaly event at these neighbor cells.
- ***Rare anomaly events***: The anomaly event labels **rarely** account for less than **0.1 percent** over all the reported KPIs. The model needs to focus on those anomaly points.



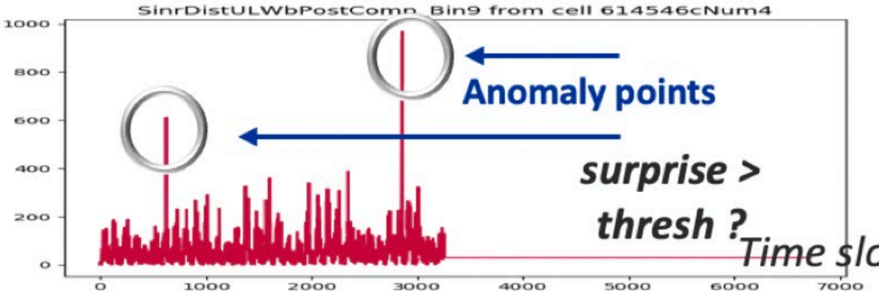
thousands of KPIs

relations?

target KPIs

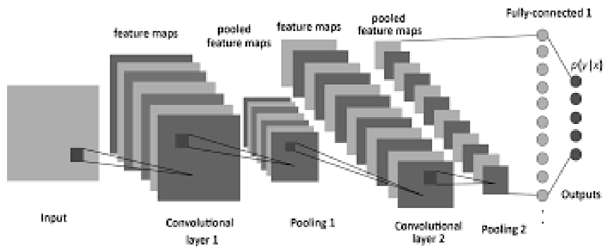


Inter-cell Interference



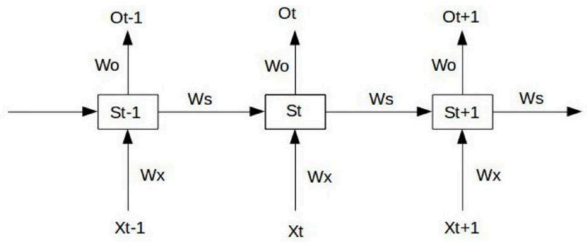
Anomaly Detection: Model Selection

CNN



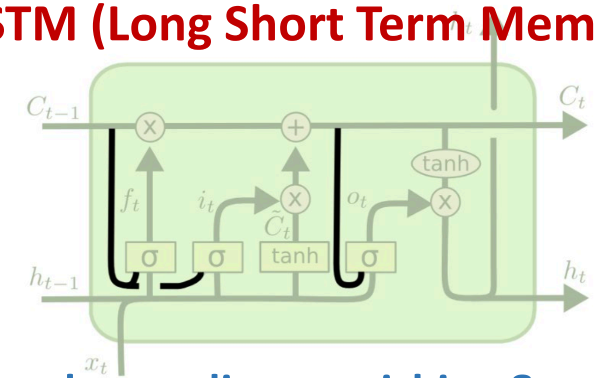
- Good at extracting spatial features from input: which KPIs are more correlated to the predictive target?
- Ignore temporal relations

RNN



- Good at extracting temporal relations between time-series inputs
- Detect "periodic" pattern
- Selectively remember "important" time slots
- Gradient vanishing & gradient explosion
- Cannot remember long-term information

LSTM (Long Short Term Memory)

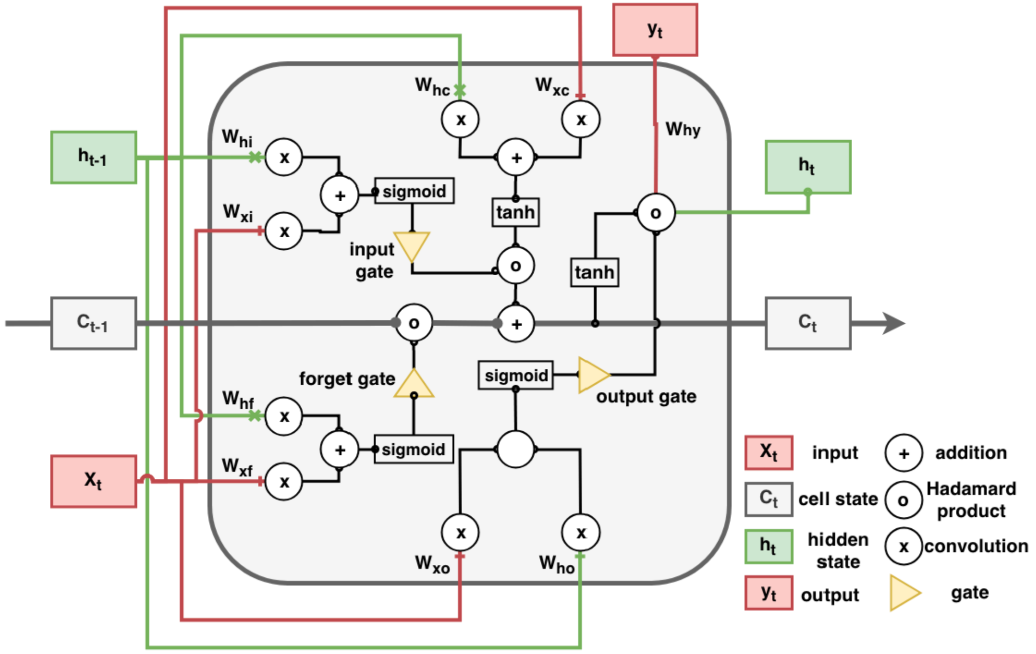


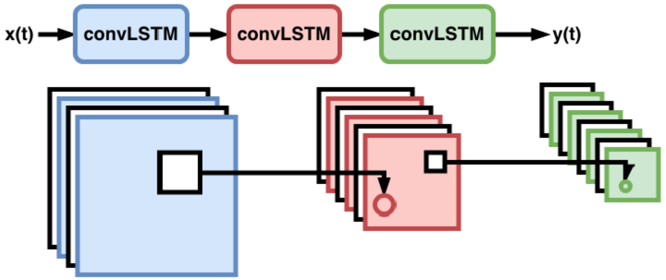
- Resolve gradient vanishing & gradient explosion
- Enable long-term memory
- Cannot well extract spatial features

RUTGERS
WINLAB | Wireless Information Network Laboratory
WINLAB

Anomaly Detection: ConvLSTM

- Extracting both ***temporal*** and ***spatial*** features





- Input: thousands of historical cell KPIs
- Output: predictive values of target cell KPIs

- Model structures (similar to *LSTM*)

input gate	$i_t = \sigma(W_{xi} * X_t + W_{hi} * H_{t-1} + W_{ci} \circ C_{t-1} + b_i$
forget gate	$f_t = \sigma(W_{xf} * X_t + W_{hf} * H_{t-1} + W_{cf} \circ C_{t-1} + b_f$
sigmoid state	$C_t^* = tanh(W_{hc^*} * H_{t-1} + W_{xc^*} * X_t + b_f)$
update gate	$C_t = f_t \circ C_{t-1} + C_t^*$
output gate	$o_t = \sigma(W_{xo} * X_t + W_{ho} * H_{t-1} + W_{co} \circ C_t + b_o)$
hidden state	$H_t = o_t \circ tanh(C_t)$
output	$Y_t = W_{hy} * H_t + b_{hy}$

- The operator "*" (convolution operations) that is the key in this model

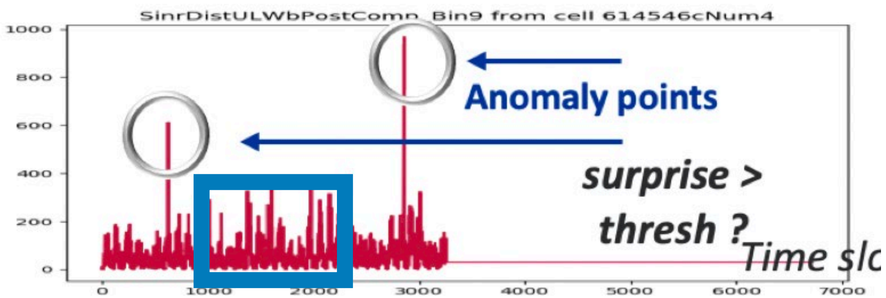
- The convolution operation enables to extract spatial features

Xingjian, S. H. I., et al. "Convolutional LSTM network: A machine learning approach for precipitation nowcasting." Advances in neural information processing systems. 2015.

Anomaly Detection: Unbalanced Dataset

How to handle extremely unbalanced dataset? (Rare anomaly events)

- ***Data undersampling***
  - ***Discard the redundant data*** that is far from the the anomaly points.



- **Penalized classification**
  - ***Penalizing error anomaly classification*** will introduce an extra cost to the model when it falsely classifies an anomaly point as a normal one. These penalties force the model to give greater emphasis to the minority class.

$$trainLoss = \alpha * normClass + \beta * \mathrm{anomalyClass}(\alpha \ll \beta)$$

RUTGERS
WINLAB | Wireless Information Network Laboratory
WINLAB

Root Cause Analysis: System Challenges

System Challenges

- Root cause labels are *not available* for supervised training
  - Network engineers do not deliberately attach the resulting fault to the associated logs
  - *Too expensive* to collect the logs by purposely introducing the cell faults

Solutions

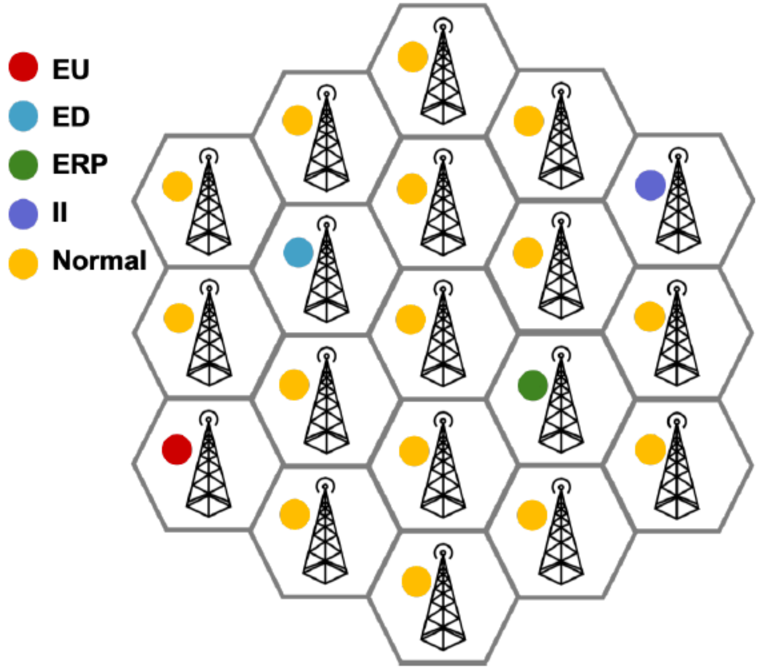
- Generate a *synthetic datase*t of cell faults with NS3
- Employ unsupervised clustering by removing the fault labels, with which we are able to quantify how the model performs
- Apply the model to a real-world dataset

RUTGERS
WINLAB | Wireless Information Network Laboratory
WINLAB

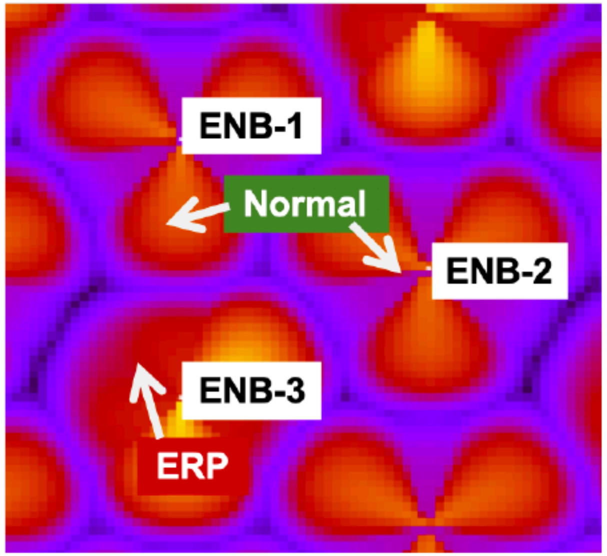
Root Cause Analysis: NS3 simulation

NS3 simulation steps

Generate “normal” topology → Randomly select x fault cells → Randomly assign fault case y → Run simulations and collect metrics (labeled by fault case) → Train/test classifier



NS3 eNB topology configuration



power radiation of normal/anomaly eNBs

RUTGERS
WINLAB | Wireless Information Network Laboratory
WINLAB

Root Cause Analysis: NS3 simulation

NS3 simulation setup

Parameter	**Value**
Topology	3-sector hexagonal grid, 3 sites
Carrier Freq.	2.12 GHz
Bandwidth	10 MHz
Channel model	UMi, shadow fading, no fast fading
TX power	**46 dBm**
Antenna	3D parabolic **70° azim.**, 10° vertical beamwidth **9° downtilt**
Handover algorithm	A3 RSRP (default Hyst = 3 dB, TTT = 256 ms)
Scheduler	Proportional fair
Mobility model	Steady state random waypoint UE speeds ∈ $U(1,20) m/s$
Traffic model	Constant bit rate 800 kbps DL + UL flows

normal cell configuration

- *EU: excessive uptilt*
- *ED: excessive downtilt*
- *ERP: excessive cell power reduction*
- *CH: coverage hole*
- *TLHO: too late handover*
- *II: inter-cell interference*

Fault Cause	**Configuration**
EU	Downtilt=[0,1] °
ED	Downtilt=[16,15,14] °
ERP	ΔP_{TX} =[7,8,9,10] dB
CH	Δ_{hole} =[49,50,52,53] dBm
TLHO	HOM=[6,7,8] dBm
II	$P_{TX_{max}}$ =33 dBm Downtilt=15 ° AB=[30, 60] ° EB=10 °
No fault	Normal

fault cell configuration

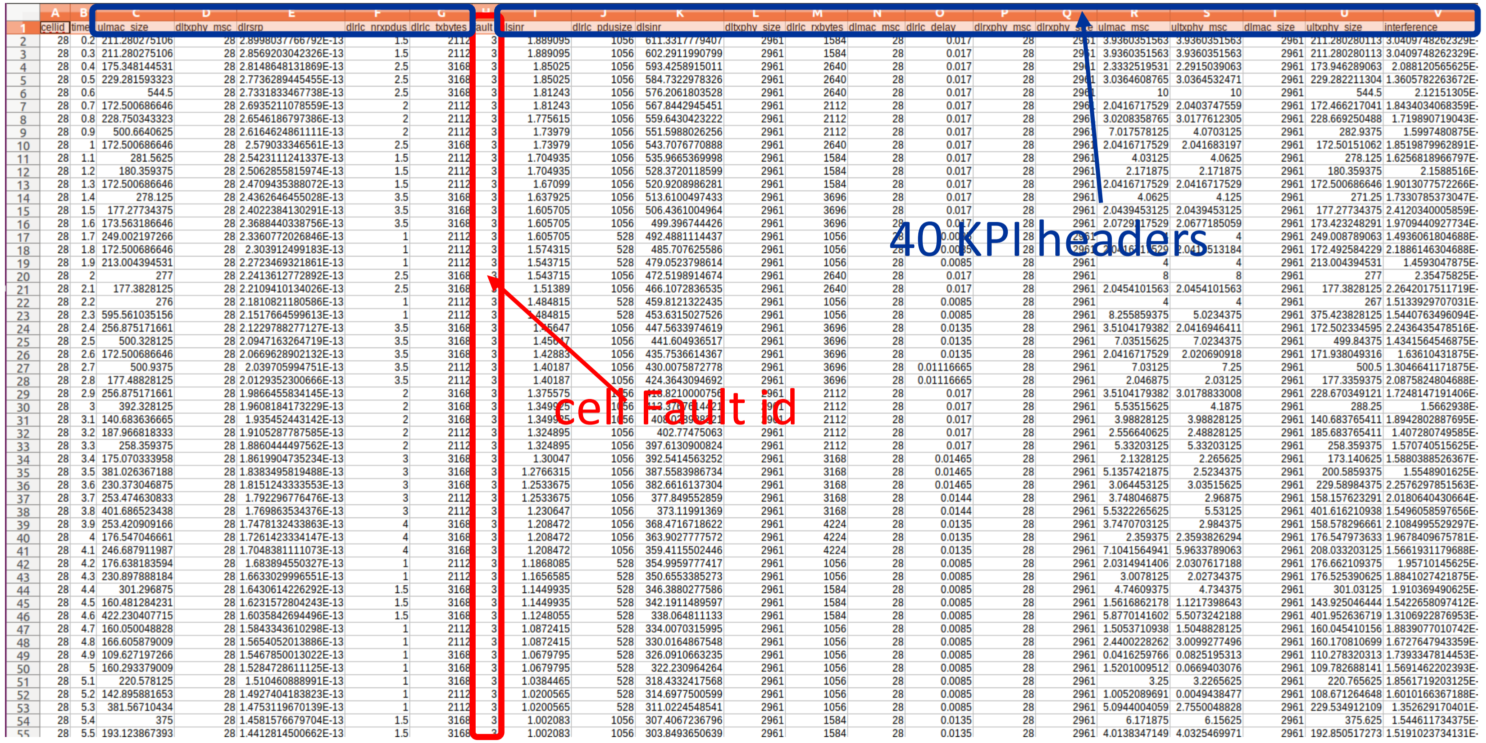
Root Cause Analysis: NS3 simulation

- **6** possible faults:
  - *EU (excessive uptilt),*
  - *ED (excessive downtilt),*
  - *ERP(excessive power reduction),*
  - *II (inter-cell interference)*
  - *TLHO (too late handover)*
  - *CH (coverage hole)*
- Randomly select **6** out of **30** cells as the faulty ones
- Randomly assign **1** possible fault to the faulty cell
- **40** KPIs

'ul_delay_max', 'ul_PduSize_avg', 'dlrx_size', 'dl_TxBytes', 'ulmac_mcs', 'dl_PduSize_std','fault', 'dl_delay_max', 'ul_delay_avg', 'ul_PduSize_min', 'ul_TxBytes', 'dltx_size', 'dl_nRxPDUs','ultx_mcs', 'ulmac_sframe', 'dlrsrp', 'ul_delay_std', 'ul_PduSize_std', 'ul_nTxPDUs', 'dist', ‘dl_PdSize_max’,‘ultx_size’,‘dl_delay_std’,‘ul_RxBytes’,‘dl_PduSize_min’, ‘dl_RxBytes’,‘ul_PdSize_max’, ‘ul_nRxPDUs’, ‘dlrx_mcs’, ‘dlsinr’, ‘dl_delay_avg’, ‘ulmac_frame’, ‘dlrx_mode’,‘dl_delay_min’, ‘ulmac_size’, ‘dl_PduSize_avg’, ‘dl_nTxPDUs’, ‘dltx_mcs’, ‘ul_delay_min’, ‘UE location’

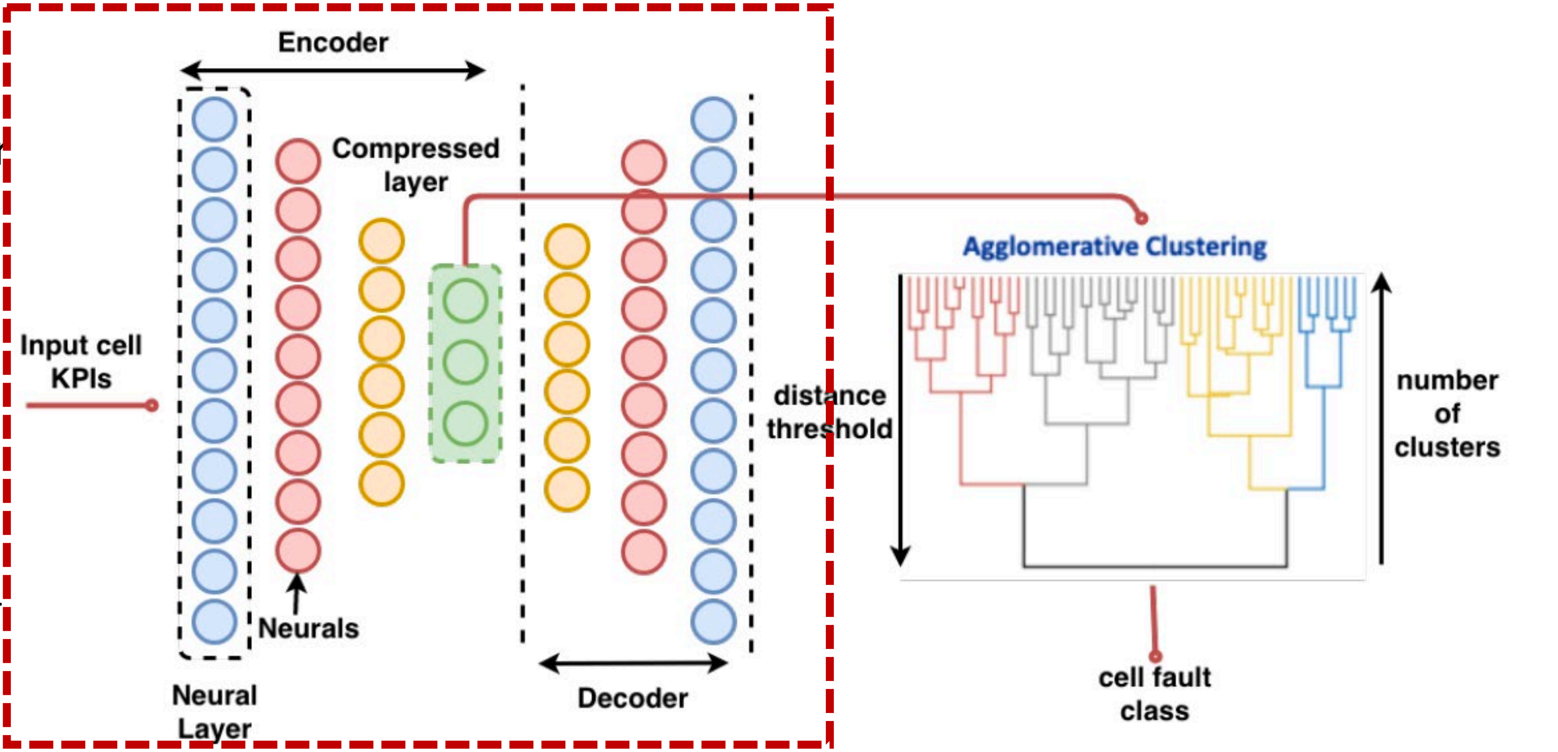
- **1** hour duration

snapshot of the dataset



Root Cause Analysis: Unsupervised Learning

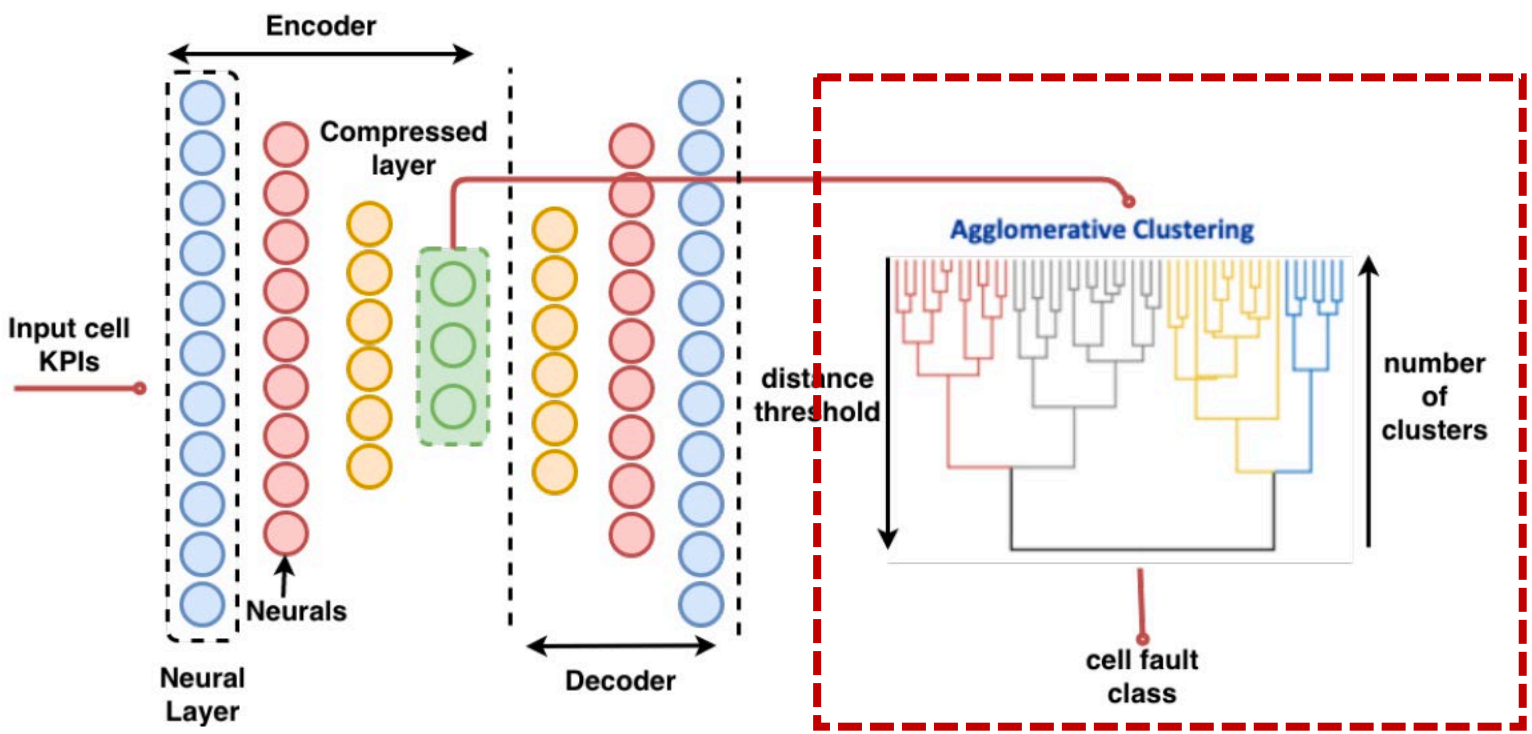
- Feature selections
  - a critical preprocessing step that ***selects a subset*** from the ***high-dimension input*** to decrease the ***overfitting*** probability and to reduce the ***training/inference time***
- Auto-encoder is an unsupervised ***data coding*** approach that can extract both linear and nonlinear relations from high-dimensional input
  - the similar feed-forward network structure with CNN and consists of two symmetrical components: encoder and decoder
    - The encoder takes the high-dimensional data and outputs the low-dimensional one, while the decoder will learn to fully recover the initial input from the compressed output with little loss.



Auto-encoder

Root Cause Analysis: Unsupervised Learning

- Agglomerative Clustering
  - A bottom-up algorithm.
  - **Flow**: starts by regarding each feature input as an independent cluster and repeats to merge two nearest clusters (measured by *Euclidean distance* or *Pearson correlation distance*) iteratively until the total remaining cluster number equals to a predefined number.
  - **Limitation**: ***cannot*** naturally map each cluster to a ***particular fault class.*** A network expert may further need to empirically infer the physical representation of each cluster, e.g., intercell interference, based on the distributions of significant KPIs.



Agglomerative Clustering

Evaluations: Anomaly Prediction

- ***Prediction Objective***: used the last 5 hours data to predict the value in the next hour of " *X2 handover failure rate*" (only an example) (using real-world dataset)

- ***Deep Learning Models*** (implemented with Tensorflow/Keras):
  - *CNN (resnet50)*
  - *LSTM*
  - *convLSTM*
  - *CNN + convLSTM*

- ***Performance Metrics***:
  - true positive (TP): the number that anomaly points are correctly predicted (key indicator)
  - false negative (FN): the number that anomaly points are missing
  - false positive (FP): the number that we give a false alarm over a a normal case
  - true negative (TN): the number that we correctly predict a normal case
  - MSE: mean square error over the anomaly points and the whole dataset

RUTGERS
WINLAB | Wireless Information Network Laboratory
WINLAB

Evaluations: Anomaly Prediction

Prediction Performance with Different ML Models

Model	TP	FP	ANOM_MSE	ALL_MSE
LSTM	1	5	0.0185	0.0041
CNN	3	11	0.032	0.0083
ConvLSTM	15	17	0.0117	0.0032
CNNConvLSTM	**18**	**23**	**0.00096**	**0.0022**

- ***ConvLSTM***, and ***CNN+convLSTM*** perform much better than *LSTM* and *CNN*
- *Important to extract spatial and temporal features at the same time*

Prediction Performance with Different Anomaly Class Weights

Weight	TP	TN	FP	FN	recall
0.01/1	**16**	**5854**	**391**	**7**	**69.5%**
0.001/1	20	4442	1802	3	86.9%
0.0001/1	23	3022	3223	0	100%

normal weight/anomaly weight

recall = TP/(TP+FN)

- An insufficiently high weight => low recall
- Excessively increase the weight => blindly classify any input as anomaly KPIs
- Needs to explore the trade-off between the anomaly prediction accuracy and the tolerance of false alarms to reach an optimal point.

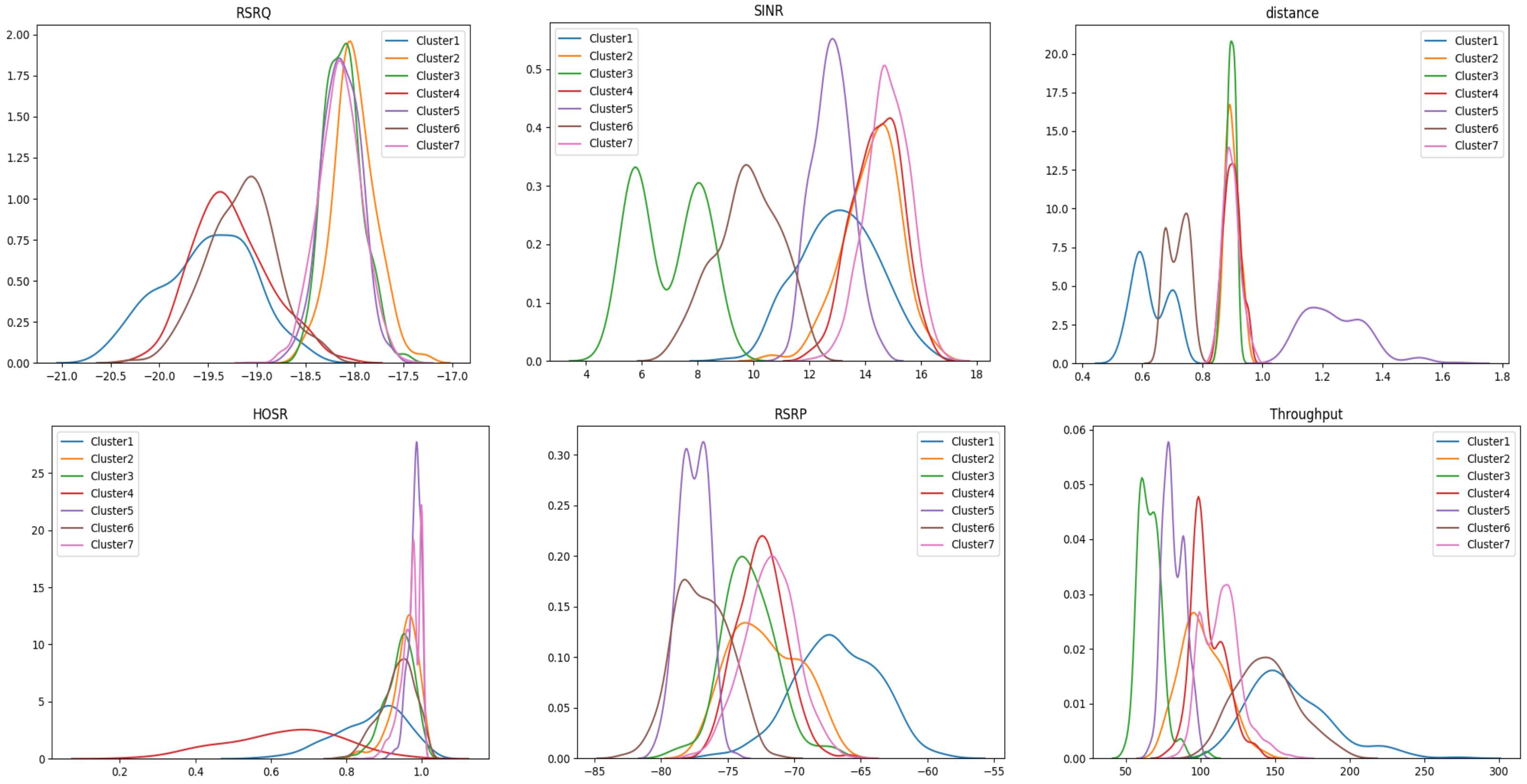
RUTGERS
WINLAB | Wireless Information Network Laboratory
WINLAB

Evaluations: Root Cause Analysis

Clustering accuracy: ***99.5 %*** by comparing the fault labels in the dataset. (***Auto-encoder + agglomerative clustering***)

TABLE VI
LABEL ASSIGNED TO EACH CLUSTER

Clusters	Label
Cluster 1	Normal
Cluster 2	Coverage Hole
Cluster 3	Too late HO
Cluster 4	Excessive reduction of cell power
Cluster 5	Excessive Uptilt
Cluster 6	Inter-system interference
Cluster 7	Excessive downtilt



KPI distributions over 6 faulty cases + 1 normal case

- Although the network cluster might be unknown, we can take it as the input to the deep reinforcement learning for the self-healing.

RUTGERS
WINLAB | Wireless Information Network Laboratory
WINLAB

Conclusions & Future Work

- Propose a ***self-organizing cellular radio access network system*** with deep learning
- Design and implement the ***anomaly prediction*** and ***root cause analysis*** components with deep learning and the evaluation of the system performance with real world data from a top-tier US cellular network operator
- Demonstrate that the proposed methods can achieve ***86.9%*** accuracy for anomaly prediction and ***99.5%*** accuracy for root cause analysis

Future Work

- Continue to design and implement the last component, "***self-healing functions***" with ***deep reinforcement learning*** and make RAN as an integrated, close-loop, self-organizing system.
- Investigate the root cause analysis with ***supervised learning*** with real-world fault labels.
- Better understand how ***KPI sampling granularity*** will effect the anomaly prediction accuracy.

RUTGERS
WINLAB | Wireless Information Network Laboratory