Self-Organizing Cellular Radio Access Network with Deep Learning

Wuyang Zhang¹, Russell Ford², Joonyoung Cho², Charlie Jianzhong Zhang², Yanyong Zhang¹,³, Dipankar Raychaudhuri¹

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/downtilt (EU/DU)
  - Coverage hole (CH)
  - Too late handover (TLHO)
  - Inter-cell Interference (II)
  - Cell Power Reduction (ERP)
  - Cell Overload (CO)

• NOT straightforward to diagnose the root cause!
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

Real-time cell KPI Monitoring

Anomaly KPI Prediction

Close Loop

Root Cause Analysis

Self Healing

S1

S2

S3

S4

Big Data Platform (Apache Spark + HDFS + Apache HBase)
Real-world KPI Dataset Overview

real-world 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: 1 hour

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

- Example KPIs in time series

- Partial example KPIs
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.
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
**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**)

\[
\begin{align*}
    i_t &= \sigma(W_{xi} * X_t + W_{hi} * H_{t-1} + W_{ci} * C_{t-1} + b_i) \\
    f_t &= \sigma(W_{xf} * X_t + W_{hf} * H_{t-1} + W_{cf} * C_{t-1} + b_f) \\
    C^*_t &= \tanh(W_{hc} * H_{t-1} + W_{xc} * X_t + b_f) \\
    C_t &= f_t * C_{t-1} + C^*_t \\
    o_t &= \sigma(W_{xo} * X_t + W_{ho} * H_{t-1} + W_{co} * C_t + b_o) \\
    H_t &= o_t \tanh(C_t) \\
    Y_t &= W_{hy} * H_t + b_{hy}
\end{align*}
\]

- The operator “*” (convolution operations) that is the key in this model
- The convolution operation enables to extract spatial features

Anomaly Detection: Unbalanced Dataset

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

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

\[
\text{trainLoss} = \alpha \ast \text{normClass} + \beta \ast \text{anomalyClass}(\alpha \ll \beta)
\]
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 dataset* 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
Root Cause Analysis: NS3 simulation

NS3 simulation steps

1. Generate "normal" topology
2. Randomly select x fault cells
3. Randomly assign fault case y
4. Run simulations and collect metrics (labeled by fault case)
5. Train/test classifier

NS3 eNB topology configuration

Power radiation of normal/anomaly eNBs
## Root Cause Analysis: NS3 simulation

### NS3 simulation setup

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Topology</td>
<td>3-sector hexagonal grid, 3 sites</td>
</tr>
<tr>
<td>Carrier Freq.</td>
<td>2.12 GHz</td>
</tr>
<tr>
<td>Bandwidth</td>
<td>10 MHz</td>
</tr>
<tr>
<td>Channel model</td>
<td>UMi, shadow fading, no fast fading</td>
</tr>
<tr>
<td>TX power</td>
<td>46 dBm</td>
</tr>
<tr>
<td>Antenna</td>
<td>3D parabolic, 70° azim., 10° vertical beamwidth, 9° downtilt</td>
</tr>
<tr>
<td>Handover</td>
<td>A3 RSRP (default Hyst = 3 dB, TTT = 256 ms)</td>
</tr>
<tr>
<td>Scheduler</td>
<td>Proportional fair</td>
</tr>
<tr>
<td>Mobility model</td>
<td>Steady state random waypoint</td>
</tr>
<tr>
<td>Traffic model</td>
<td>Constant bit rate 800 kbps DL + UL flows</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Fault</th>
<th>Cause</th>
<th>Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>EU</td>
<td></td>
<td>Downtilt=[0,1] °</td>
</tr>
<tr>
<td>ED</td>
<td></td>
<td>Downtilt=[16,15,14] °</td>
</tr>
<tr>
<td>ERP</td>
<td></td>
<td>$\Delta P_{T,X} = [7,8,9,10]$ dB</td>
</tr>
<tr>
<td>CH</td>
<td></td>
<td>$\Delta h_{ole} = [49,50,52,53]$ dBm</td>
</tr>
<tr>
<td>TLHO</td>
<td></td>
<td>HOM=[6,7,8] dBm</td>
</tr>
<tr>
<td>II</td>
<td></td>
<td>$P_{T,X_{max}} = 33$ dBm</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Downilt=15 °</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AB=[30, 60] °</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EB=10 °</td>
</tr>
<tr>
<td></td>
<td>No fault</td>
<td>Normal</td>
</tr>
</tbody>
</table>

### Traffic model

- Normal cell configuration
- 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','ulRxBytes','dl_Pd
  - uSize_min', 'dl_RxBytes', 'ul_PdSize_max', 'ul_nRxPDUs',
  - 'dirx_mcs', 'dlsinr', 'dl_delay_avg', 'ulmac_frame',
  - 'dirx_mode', 'dl_delay_min', 'ulmac_size', 'dl_PduSize_avg',
  - 'dl_nTxPDUs', 'dltx_mcs', 'ul_delay_min', 'UE location'

- 1 hour duration
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.
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.
• **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 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
## Evaluations: Anomaly Prediction

### Prediction Performance with Different ML Models

<table>
<thead>
<tr>
<th>Model</th>
<th>TP</th>
<th>FP</th>
<th>ANOM_MSE</th>
<th>ALL_MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSTM</td>
<td>1</td>
<td>5</td>
<td>0.0185</td>
<td>0.0041</td>
</tr>
<tr>
<td>CNN</td>
<td>3</td>
<td>11</td>
<td>0.032</td>
<td>0.0083</td>
</tr>
<tr>
<td>ConvLSTM</td>
<td>15</td>
<td>17</td>
<td>0.0117</td>
<td>0.0032</td>
</tr>
<tr>
<td>CNNConvLSTM</td>
<td>18</td>
<td>23</td>
<td>0.00096</td>
<td>0.0022</td>
</tr>
</tbody>
</table>

- **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

<table>
<thead>
<tr>
<th>Weight</th>
<th>TP</th>
<th>TN</th>
<th>FP</th>
<th>FN</th>
<th>recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.01/1</td>
<td>16</td>
<td>5854</td>
<td>391</td>
<td>7</td>
<td>69.5%</td>
</tr>
<tr>
<td>0.001/1</td>
<td>20</td>
<td>4442</td>
<td>1802</td>
<td>3</td>
<td>86.9%</td>
</tr>
<tr>
<td>0.0001/1</td>
<td>23</td>
<td>3022</td>
<td>3223</td>
<td>0</td>
<td>100%</td>
</tr>
</tbody>
</table>

- 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.

recall = TP/(TP+FN)
Evaluations: Root Cause Analysis

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

<table>
<thead>
<tr>
<th>Clusters</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cluster 1</td>
<td>Normal</td>
</tr>
<tr>
<td>Cluster 2</td>
<td>Coverage Hole</td>
</tr>
<tr>
<td>Cluster 3</td>
<td>Too late HO</td>
</tr>
<tr>
<td>Cluster 4</td>
<td>Excessive reduction of cell power</td>
</tr>
<tr>
<td>Cluster 5</td>
<td>Excessive Uptilt</td>
</tr>
<tr>
<td>Cluster 6</td>
<td>Inter-system interference</td>
</tr>
<tr>
<td>Cluster 7</td>
<td>Excessive downilt</td>
</tr>
</tbody>
</table>

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.
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.