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], and Dipankar Raychaudhuri[1]

[1]WINLAB, Rutgers University
[2]Samsung Research America
[3]University of Science and Technology of China
Contact: *wuyang*@winlab.rutgers.edu, russell.ford@samsung.com

***Abstract*— Network operators require a high level of performance and reliability for the cellular radio access network (RAN) to deliver high quality of service for mobile users. However, these network do experience some rare and hard-to-predict anomaly events, for example, hardware failures and high radio interference, which can significantly degrade performance and end-user experience. In this work, we propose SORA, a self-organizing cellular radio access network system enhanced with deep learning. SORA involves four core components: self-KPIs monitoring, self-anomaly prediction, self-root cause analysis, and self-healing. In particular, we design and implement the anomaly prediction and root cause analysis components with deep learning methods and evaluate the system performance with 6 months of real-world data from a top-tier US cellular network operator. We demonstrate that the proposed methods can achieve 86.9% accuracy in predicting anomalies and 99.5% accuracy for root cause analysis of network faults.**

I. Introduction

The performance and reliability of the cellular radio access network (RAN) heavily impacts the quality of service of mobile users. However, the RAN can occasionally suffer from anomaly events or faults, e.g., excessive antenna uptilt/downtilt, too late handover, and inter-cell interference, in addition to random hardware and software failures, that may significantly degrade performance and user experience. These degradation events can be hard to identify and predict through manual analysis by field engineers.

To maintain the performance of RAN, hundreds of key performance indicators (KPIs) from thousands of cellular base stations must be continuously monitored. Once any anomaly KPIs appear, network engineers may spend significant time and effort to manually perform root cause analysis (RCA), i.e., inferring the possible causes for the degraded performance, after which the problem can be remedied. This effort requires expert domain knowledge and can take hours or even days to diagnose the problem.

By the estimates in [1], network management costs account for 10-30% of operator revenue, which can total in the tens of billions for major telcos. Motivated by minimizing these high OpEx costs, in this paper we consider how network operators can reduce or eliminate manual network configuration and maintenance through automated monitoring, diagnosis and even self-healing of the LTE RAN. The benefits of automation are clear from the perspectives of better end-user experience (fewer dropped calls, improved call setup success rates, higher end-user throughput, alleviated congestion, higher subscriber satisfaction and loyalty) and operational efficiencies (accelerated rollout times, automated network upgrades, energy and cost savings, and reduced labor fees for network maintenance and diagnosis) [2], [3].

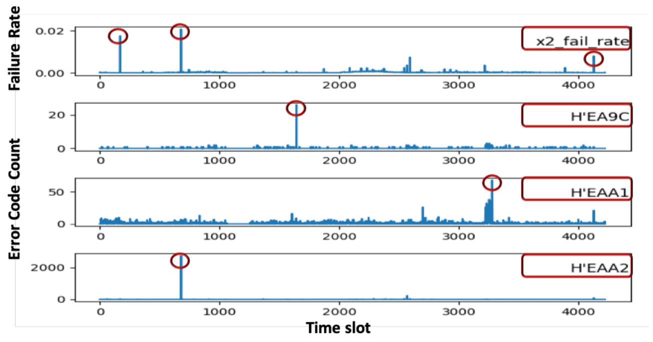


Fig. 1. Example real-world KPIs and error codes. The first sub-figure shows LTE x2 handover failure rate and the last three sub-figures presents three error codes that specify particular error cases. Importantly, we highlight the rare anomaly points.

In order to reach this goal, several aspects of self-organizing networks have been studied in recent years. For example, The work in [4] employs decision trees for analyzing call drops and achieves an accuracy of 92.1% in diagnosing the root causes of drops. AT&T [5] proposed G-RCA, a root cause analysis system that ensures service quality management in IP networks. The work in [6] discussed a self-organizing map (SOM) approach to perform unsupervised training to automatically identify LTE faults. Network anomaly detection proposed in [7] proactively profiles IP traffic patterns to automate network management.

Despite the earlier and ongoing work on various aspects of network anomaly detection and root cause analysis, the problem of how to efficiently construct a closed-loop self-organizing system, especially exploiting deep learning, has not been systematically studied. Moreover, little previous work studies how to perform network anomaly prediction before any anomaly events actually occur, which results in revenue loss for network operators.

In this work, we propose an automated system "SORA" that intends to fully enable a self-organizing cellular radio access network. SORA, as a self closed-loop system, includes

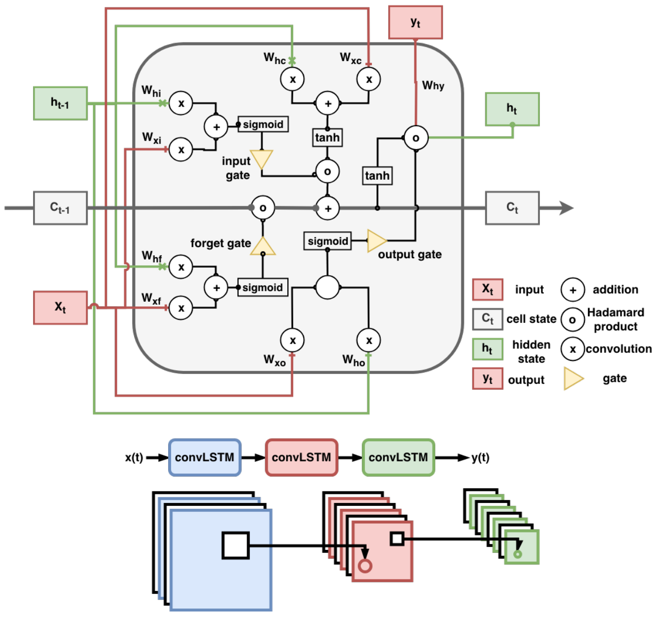


Fig. 2. The network structure of convLSTM

four core components: (1) the real-time KPIs monitoring function that reports thousands of cell KPIs to a local database or aggregated values to a data center in a timely fashion; (2) the self anomaly KPIs/event prediction function that forecasts anomalies in advance; (3) self root cause analysis that diagnose the root reason of a particular anomaly case; and (4) the self-healing function that automatically recovers from any specific fault.

In this paper, the design and implementation of SORA mainly focuses on: (1) How can we accurately predict anomaly KPIs that may further lead to faulty operation before they appear? (2) How to automatically identify the root causes of faults from thousands cell KPIs and error codes? We leave the implementation of the self-healing function in our near future work but discuss the design plan in Section II. Also, we collaborate with a major vendor of LTE base station equipment and make use of field data from a top-tier US cellular operator to perform the system design and the performance evaluation. Due to the challenges of obtaining the labeled real-world data needed for supervised learning, we use NS3 simulation to drive the evaluation for our RCA investigation.

We summarize our contributions in this work as follows.

- We have proposed a self-organizing cellular radio access network system with deep learning and have implemented two core components: prediction for anomaly cell KPIs/events and root cause analysis.
- We performed anomaly cell KPIs/events prediction with CNN+convLSTM and achieved an 86.9% precision trained and tested against 6 months of aggregated, real-world operator data.
- We performed root cause analysis with both supervised classification (auto-encoder+decision tree) and unsupervised clustering (auto-encoder+agglomerative clustering), achieving 99.5% accuracy.

II. System Design

In this section, we first propose a self-organizing cellular RAN system SORA with deep learning and then discuss the design of two core components: anomaly prediction and root cause analysis.

Here, we introduce the four-step closed-loop of SORA: (1) **Real-time KPIs monitoring**. Each individual cell reports thousand KPIs, for example, *'Average UE Throughput in DL/UL', 'Packet Drop Rate', 'Total UL Interference Power', 'Average DL/UL CQI', etc*, in addition to error codes indicating reasons for call drops or other control-plane faults. In particular, we divide all those KPIs into 7 groups that exhibit the system status from the distinct aspects: Accessibility, Retainability, Integrity, Availablity, Mobility, Connection Drop Rate, and Cell Throughput. Those KPIs can be either streamed over the network to a remote data center or stored in local edge clouds and monitored by the network in real-time. Example real-world KPIs and error codes are presented in Figure 1. (2) **Anomaly KPIs/events prediction**. Any anomaly KPIs indicate that a cell is somehow operating outside of ideal limits. Some important KPI anomalies may indicate some critical fault has occurred, causing significant performance degradation. In order to prevent the system from falling into those states, SORA needs to predict, based on the current status of the collected KPIs, the variation tendency of the critical KPIs. SORA acquires a few numbers of the critical KPIs from the input of a network expert. When the prediction model, presented in Section II-A, detects any value of those critical KPIs exceeding a predefined threshold (we consider the case that the value goes beyond 3 times standard deviation from the historical values in this work), SORA will mark the anomaly point and then pass this input to the next step, automated root cause analysis. Otherwise, SORA returns to the first step and keeps the loop running. (3) **Root cause analysis**. In this step, SORA attempts to diagnose the root causes that result in the anomaly KPIs detected in the previous step. Example root causes are excessive antenna uptilt/downtilt, excessive power reduction, coverage hole, too late handover, etc. Accurately identifying the root cause is the key to the self-organizing system. See II-B. (4) **System self-healing**. Finally, SORA can self cover from the fault detected from the last step by changing the associated parameters. As shown in [8], deep reinforcement learning can be a candidate tool that greatly helps for this step. It enforces the model to learn optimal actions based on the observed environment. We leave the implementation of this step to our future work.

Moreover, the system will need to cover the service of many LTE eNodeB base station site, which for some operators can number in the tens of thousands. Therefore, a big data platform is required for the storage and computation to handle the real-time, closed-loop system. In this work, we base our big data platform on scalable software such as Apache HDFS, HBase and Spark [9]. However, the full details of our implementation are outside the scope of this paper.

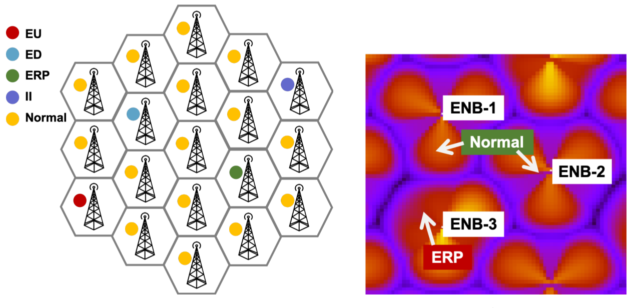
A. Anomaly KPIs/Events Prediction

Objective and System Challenges. Our objective is to predict, based on the currently-reported cell KPIs, the potential anomaly KPIs/events in the future. This is a challenging problem because of the following reasons.

1) It may be difficult to know in advance which of the thousands of KPIs (from the same or nearby cells) are relevant and correlated with the anomaly KPIs.
2) 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. Therefore, the analysis needs to extract both temporal and spatial characteristics from the data collected in a multi-cell environment.
3) The anomaly event labels rarely account for less than 0.1 percent over all the reported KPIs. Thus, a predictive model may learn to ignore anomaly points and to classify any point as a normal case because it will still achieve a high predictive result without needing to capture the anomaly pattern. Therefore, we have to take care that the model gives greater emphasis to these rare but critical anomaly points.

To address the first two challenges, we need to exploit a neural network model that is able to simultaneously extract temporal and spatial features for anomaly prediction. Convolution neural network (CNN) [10] is good at extracting spatial features and accordingly selecting those significant ones from a large volume of features. Therefore, CNN will be useful for extracting the features (KPIs and error codes) that highly associate with a predictive target, while it is not well-suited to explore the time dependency within time-series data. Meanwhile, recurrent neural networks (RNNs) [11] have been proposed to extract temporal relations between time-series input and to search periodic patterns by selectively remembering the "important" time slots. In particular, a Long Short-Term Memory (LSTM) [12] network is popular RNN variant, which resolves the exploding/vanishing gradient problem of standard RNNs. Still, while CNN and LSTM individually have many merits, they do not completely cover our requirements.

ConvLSTM to for spatiotemporal feature extraction. Our anomaly prediction work is makes use of the convLSTM model, proposed in [13], which combines the benefits of both CNN and LSTM to handle the spatiotemporal prediction problems. It retains the fundamental structure with LSTM, while converts all vector products to convolutional operations. This variation efficiently enables the model to extract spatial features from multiple LTE KPIs at every time slot. Here, we summarize how convLSTM works. It first takes X(t), which is a vector containing all the KPIs in the last k time slots concatenated together. In the dataset used in our evaluation in Section III-A, data is aggregated into 1 hour time slots. A smaller granularity of time slot would likely provide better predictions but also introduce higher load to store and compute data. The value of k is a tuning parameter in the network and we set it to 5 in our



(a) (b)

Fig. 3. (a) NS3 eNB topology configuration; (b) power radiation of normal/anomaly eNBs.

case as it provides the highest prediction accuracy. Next, multiple convLSTM layers, as shown in Figure 2, will learn the important features that contribute most to the predictive target at each time slot from all input features as well as remember the long-term temporal features. We present the detail of a single convLSTM layer in Figure 2 and summarize how to calculate the input at each layer in Equation 1. The network is trained to perform regression: the output layer is a single neuron with a linear activation function and outputs the predicted real value of the anomaly KPI of interest. Note the operator "$*$" is the convolution operation that is the key in this model and "$\circ$" denotes the Hadamard production. We further evaluate the performance of convLSTM and also compare it with CNN and LSTM in Section III-A.

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

To address the third challenge of extremely unbalanced label classes, where the anomaly labels in the historical data rarely make up 0.1 percent, we consider two strategies to achieve a higher prediction accuracy. (1) **Dataset under-sampling**. Under-sampling is an efficient data preprocessing method when the dataset is extremely unbalanced. Since the desired objective is to accurately predict those anomaly KPIs, we can easily discard the redundant dataset that is far from the time when the anomaly points appear. After removing those inconsequential points, the model can concentrate on the points surrounding the anomaly points and explore the pattern of how KPIs will distribute before an anomaly appear. (2) **Penalized classification**. Penalizing fault 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. Here, we adjust the weights for the anomaly class and the normal class as shown in Equation 2. The overall loss value during the training, therefore, will significantly decrease when the model correctly predicts an anomaly point but less so when it successfully identifies a normal point. We will evaluate the predictive performance by changing the weight ratio in Section III-A.

$$training\_loss = \alpha * normal\_class + \beta * anomaly\_class \quad (2)$$

B. Root Cause Analysis

System Challenges. After detecting anomaly points, we need to determine the root cause. Here, the system challenge for machine learning is that those root cause labels are not always available because detailed logs of historical network faults are often unavailable to label the corresponding KPI data (the ground truth). This challenge mainly stems from the fact that network engineers did not deliberately attach the resulting fault to the associated logs and also it is too expensive to collect the logs by purposely introducing the cell faults. Thus we cannot easily train a supervised learning model with the real-world data of cell faults. To address this problem, we first generate a synthetic dataset of cell faults with NS-3 [14], a popular discrete-event network simulator for Internet systems, and apply supervised classification over the dataset. Next, we employ unsupervised clustering afterward on the generated dataset by removing the fault labels with which we are able to quantify how the model performs. With a reasonably accurate simulation model, we can finally apply it to a real-world dataset.

Generating cell faults with NS3. To get the label data for supervised learning, we use NS-3 simulation to generate the cell KPIs along with fault labels. We first configure the normal LTE environment by taking the associated parameters from the work [6] (see Table II-B). Then we randomly select 20% out of total N cells and randomly assign one of the following faults to each cell: *excessive uptilt/downtilt (EU/DU), coverage hole (CH), too late handover (TLHO), inter-cell interference (II), excessive cell power reduction (ERP) and cell overload (CO)*. To introduce a particular cell fault, we modify the parameters as shown in Table II, also following [6]. An example LTE network topology and cell radiation are shown in Figure 3. Finally, we run 64 groups 20-minute simulation to collect 40 KPIs and fault labels from all N cells.

Feature selection with auto-encoder. Feature selection is 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 [15] is an unsupervised data coding approach that can extract both linear and nonlinear relations from high-dimensional input. It shares 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. After training the auto-encoder with the NS-3 dataset generated in the previous step, we take the encoder as a feature selection tool to efficiently yield a low-dimensional input.

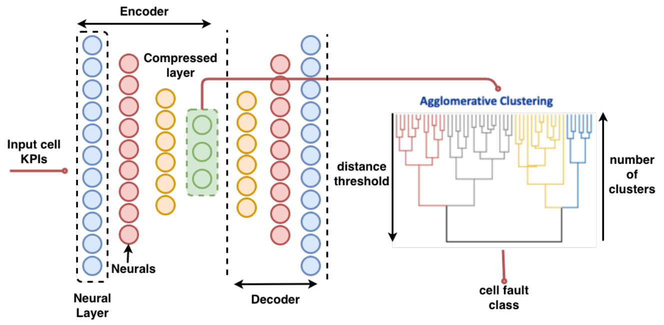


Fig. 4. Unsupervised learning with agglomerative clustering.

Supervised classification with decision tree. With the compressed output from the last step, we use a decision tree to perform supervised cell fault classification. A decision tree consists of a binary tree structure of decision nodes that compare the input value with its threshold and accordingly steer it to the right/left child node. To construct the tree and decide each threshold, the decision tree takes a recursively top-down approach to minimize the entropy of all the nodes. In our system, a decision tree takes the selected features from the auto-encoder and classifies them as one of the possible cell faults or a normal case.

Unsupervised learning with agglomerative clustering. We employ an unsupervised learning approach that takes the compressed data from the auto-encoder but runs the training process without any cell fault labels. Agglomerative clustering [16] is a bottom-up algorithm as shown in Figure 4. It 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. The limitation of this algorithm is that we cannot naturally map each cluster to a particular fault class and thus a network expert may further need to empirically infer the physical representation of each cluster, e.g., inter-cell interference, based on the distributions of significant KPIs, as discussed in [6]. More details are given in III-A. Nevertheless, this cluster-to-fault class mapping is not always necessary since the last step, "self-healing", in SORA is to take a particular cluster even without a specific class mapping and to make, with deep reinforcement learning, a proper operation to get the LTE system back to a normal status.

III. Performance Evaluation

In this section, we first compare the accuracy of anomaly KPIs/events prediction with 4 different deep learning models and conclude that CNN + convLSTM delivers the best performance with 86.9% accuracy. Then we show that our supervised and unsupervised learning approaches can perform root cause analysis in the radio access network with an accuracy of 99.87% and 99.5%, respectively.

A. Anomaly Prediction

We use the last 5 1-hour slots of aggregated KPIs to predict the value in the next hour and take "X2 handover failure rate" as an example predictive target to drive our evaluation. We implemented 4 deep learning models with Keras/Tensorflow: LSTM, CNN, convLSTM and CNN + convLSTM.

Performance metrics. To evaluate the prediction accuracy, we consider a confusion matrix including 4 metrics: true positive (TP), false negative (FN), false positive (FP) and true negative (TN). In particular, TP counts the number that we predict anomaly values correctly, while FN is the number that we fail to predict anomaly values. FP means that we give a false alarm over a normal case. Finally, TN says that we predict a normal case correctly. The objective is to maximize TP and to minimize FP. Moreover, we consider the mean squared error (MSE) between the predictive values and the ground truth upon both the anomaly points (ANOM_MSE) and the overall dataset (OVERALL_MSE).

Table III summarizes the evaluation results of 4 different deep learning models. We find CNN + convLSTM presents the best performance by successfully detecting the maximal number of anomaly points and also giving the lowest ANOM_MSE and OVERALL_MSE. Importantly, we find the last two methods involving convLSTM significantly outperform than LSTM and CNN. This fact indicates that considering either spatial features with CNN or temporal features with LSTM alone will worsen the performance, comparing to the case of involving both of them. We conclude that this method efficiently extracts key features from high-dimensional KPIs with CNN at the beginning and then forwards the intermediate results to convLSTM that analyzes spatiotemporal features.

Moreover, as we discussed in Section II-A, deliberately adding a higher weight to the anomaly class during the training can enforce the model to explore which KPIs variation pattern will lead to anomaly points. We test 3 different class weights between non-anomaly class and anomaly class: $0.01/1, 0.001/1, 0.0001/1$ and present the evaluation result in Table IV. We notice that assigning an insufficiently high weight, e.g., $0.01/1$ to the anomaly class only has a 69.5% recall. However, if we excessively increase the weight to a large value, say $0.0001/1$, the model will blindly classify any input as an anomaly which meanwhile introduces a large volume of false alarms. Finally, we adjust the weight to an intermediate value, e.g., $0.001/1$. In this case, the model outputs an 86.9% and cuts down the false alarms number (FP) into half compared to the second setting (see Table III). We conclude that an application using this algorithm needs to explore the trade-off between the anomaly prediction accuracy and the tolerance of false alarms to reach an optimal point.

B. Root Cause Analysis

We now demonstrate how effective the proposed approaches (both supervised classification and unsupervised clustering) can be for root cause analysis. Note that we use the labeled data generated by NS3 to drive the evaluation for both approaches. The performance metric is *accuracy* that measures the gap between the predicted class and the actual labeled class. First, we review the supervised classification with decision tree without auto-encoder. A decision tree usually presents a higher classification accuracy with a more complicated model from a deeper tree. Meanwhile, a deeper tree introduces higher computation overheads and increases the chance of over-fitting. Thus, we look for a balance between the classification accuracy and the tree depth. Here, we shift the tree depth from 3 to 49 and the overall classification accuracy of all 5 fault cases converges to 99.87% after increasing the depth to 26 as shown in Figure 5 (a). In addition, Figure 5 (b) shows that the distribution of correct/false classification of 5 fault cases. We notice the

TABLE I
CELL CONFIGURATIONS.

Parameters	Values
Topology Setting	*3-sector hexagonal grid, 3 sites*
Carrier Frequency	*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 = 3dB, TTT=256ms)*
Scheduler	*proportional fair*
Mobility Model	*steady state random waypoint, UE speed: (1,20)m/s*
Traffic Model	*constant bit rate 800 kbps DL + UL flows*

TABLE II
CELL FAULT CONFIGURATIONS

Fault Cases	Descriptions	Configurations
EU	Excessive Antenna Uptilt	Downtilt=[0,1]°
ED	Excessive Antenna Downtilt	*Downtilt=[16,15,14]°*
ERP	Excessive Power Reduction	ΔP_{TX}*=[7,8,9,10] dB*
CH	Coverage Hole	Δ*hole=[49,50,52,53] dB*
TLHO	Too Late Handover	*HOM=[6,7,8] dBm*
II	Inter-cell Interference	$P_{TX_{max}}$*=33dBm* *downtilt=15°* *EB=10°*
No Fault	eNB in Normal State	*No Change*

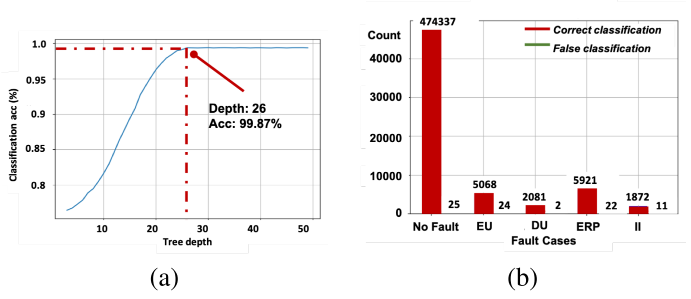


Fig. 5. (a) the root cause classification accuracy by changing the decision tree depth; (b) the classification results distribution for each fault case.

TABLE III
ANOMALY PREDICTION WITH DIFFERENT DEEP LEARNING MODELS

Model	TP	ANOM_MSE	OVERALL_MSE
LSTM [12]	1	0.0185	0.0041
CNN [10]	3	0.032	0.0083
ConvLSTM	15	0.0117	0.0032
CNN+ConvLSTM	18	0.00096	0.0022

TABLE IV
ConvLSTM with Different Class Weights

Class 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	100.0%

decision tree presents equal performance among different cell faults without showing any bias. Next, We add an auto-encoder in front of the decision because it can greatly reduce the input dimension and will be necessary to handle thousands of input KPIs in real-time root cause analysis. In this setting, we find that the highest accuracy goes down to 98%. because the information loss as the auto-encoder compresses input KPIs.

Next, we evaluate unsupervised clustering with agglomerative clustering. We assume that the labeled fault class is unknown in this approach but only use it for the evaluation purpose. We consider 7 critical cell KPIs: *RSRQ (Reference Signal Received Quality), SINR (Signal-to-noise-plus-interference Ratio), Distance (Distance between UE and its connected eNB), Retainability, HOSR (Handover Success Rate), RSRP (Reference Signal Received Power), Cell Throughput* and 6 faults as shown in Table II. We find that agglomerative clustering shows a 99.5% accuracy. Thus, our approach can accurately classify errors by each category. Importantly, in a real-world setting, without the labeled data, each cluster will not be associated with a specific fault class and a network expert needs to infer the resulting class based on the KPIs distributions of each cluster, methodology for which is presented in [6].

IV. Conclusion and Future Work

In this paper, we propose SORA, a self-organizing cellular radio access network system with deep learning. SORA involves four core components: self-KPIs monitoring functions, self-anomaly prediction functions, self-root cause analysis and self-healing. The main contribution of this work are the design and implementation of the anomaly prediction and root cause analysis components with deep learning methods and the evaluation of the system performance with real-world data from a top-tier US cellular network operator. We demonstrate that the proposed methods can achieve 86.9% accuracy for anomaly prediction and 99.5% accuracy for root cause analysis. Moving forward, we will continue to design and implement the last component, "self-healing functions" with deep reinforcement learning and make the LTE network as an integrated, close-loop, self-organizing system. We will also investigate the root cause analysis with supervised learning with real-world fault labels. Another future work topic is that of better understanding how sampling granularity will effect the anomaly prediction accuracy.

References

[1] Martha DeGrasse, RCRWireless, “Reducing Network Opex through Infrastructure: Drones, C-RAN, software and more.” `https://www.strategicvenue.com/wp-content/uploads/2017/09/RCRWireless_September2017_Opex.pdf`.

[2] “Self-Organizing Networks in the 5G Era.” `https://www.researchandmarkets.com/reports/4621704/son-self-organizing-networks-in-the-5g-era`.

[3] X. Fan, H. Ding, S. Li, M. Sanzari, Y. Zhang, W. Trappe, Z. Han, and R. E. Howard, “Energy-ball: Wireless power transfer for batteryless internet of things through distributed beamforming,” *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 2, no. 2, p. 65, 2018.

[4] A. P. Iyer, L. E. Li, and I. Stoica, “Automating diagnosis of cellular radio access network problems,” in *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking*, pp. 79–87, ACM, 2017.

[5] H. Yan, L. Breslau, Z. Ge, D. Massey, D. Pei, and J. Yates, “G-rca: a generic root cause analysis platform for service quality management in large ip networks,” *IEEE/ACM Transactions on Networking (TON)*, vol. 20, no. 6, pp. 1734–1747, 2012.

[6] A. Gómez-Andrades, P. Muñoz, I. Serrano, and R. Barco, “Automatic root cause analysis for lte networks based on unsupervised techniques,” *IEEE Transactions on Vehicular Technology*, vol. 65, no. 4, pp. 2369–2386, 2016.

[7] G. Fernandes Jr, L. F. Carvalho, J. J. Rodrigues, and M. L. Proença Jr, “Network anomaly detection using ip flows with principal component analysis and ant colony optimization,” *Journal of Network and Computer Applications*, vol. 64, pp. 1–11, 2016.

[8] Y. Duan, X. Chen, R. Houthooft, J. Schulman, and P. Abbeel, “Benchmarking deep reinforcement learning for continuous control,” in *International Conference on Machine Learning*, pp. 1329–1338, 2016.

[9] M. Zaharia, R. S. Xin, P. Wendell, T. Das, M. Armbrust, A. Dave, X. Meng, J. Rosen, S. Venkataraman, M. J. Franklin, *et al.*, “Apache spark: a unified engine for big data processing,” *Communications of the ACM*, vol. 59, no. 11, pp. 56–65, 2016.

[10] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *nature*, vol. 521, no. 7553, p. 436, 2015.

[11] A. Graves, A.-r. Mohamed, and G. Hinton, “Speech recognition with deep recurrent neural networks,” in *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*, pp. 6645–6649, IEEE, 2013.

[12] F. A. Gers, J. Schmidhuber, and F. Cummins, “Learning to forget: Continual prediction with lstm,” 1999.

[13] Z. Yuan, X. Zhou, and T. Yang, “Hetero-convlstm: A deep learning approach to traffic accident prediction on heterogeneous spatio-temporal data,” in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 984–992, ACM, 2018.

[14] R. Ford, M. Zhang, S. Dutta, M. Mezzavilla, S. Rangan, and M. Zorzi, “A framework for end-to-end evaluation of 5g mmwave cellular networks in ns-3,” in *Proceedings of the Workshop on ns-3*, pp. 85–92, ACM, 2016.

[15] J. Deng, Z. Zhang, F. Eyben, and B. Schuller, “Autoencoder-based unsupervised domain adaptation for speech emotion recognition,” *IEEE Signal Processing Letters*, vol. 21, no. 9, pp. 1068–1072, 2014.

[16] D. Beeferman and A. Berger, “Agglomerative clustering of a search engine query log,” in *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 407–416, ACM, 2000.