Machine Learning Approach for Automatic Fault Detection and Diagnosis in Cellular Networks

Jamale Benitez Porch*, Chuan Heng Foh†, Hasan Farooq* and Ali Imran*
*School of Electrical and Computer Engineering, University of Oklahoma, Tulsa, USA
{jaybp95@gmail.com, ali.imran@ou.edu, hasan.farooq@ou.edu}
†5G Innovation Centre (5GIC), Institute for Communication Systems
University of Surrey, Guildford GU2 7XH, UK
{c.foh@surrey.ac.uk}

***Abstract*—The capability for a network to self heal itself is a promising feature for future cellular networks. An essential function to achieve self healing is the ability to determine when a network is operating outside of normal state, and perhaps identify potential causes. This paper focuses on applying the supervised machine learning approach to detect fault symptoms and identify the cause. Our method utilizes referenced signal received power (RSRP) reported by users over a certain period of time to detect operational anomaly in a base station. We notice that certain faults at a base station create noticeable change in the RSRP readings and recognizable electromagnetic radiation pattern around the base station. To achieve fault analysis, we develop a framework that differentiates normal and abnormal operations under changing environment to avoid unnecessary fault alarms. Once abnormal operation is detected, the framework uses a supervised machine learning system to classify the detected fault. We develop convolutional neural network and random forest to test the fault classification. We show that both machine learning systems offer high accuracy.**

***Index Terms*—Machine learning; cellular network;**

I. Introduction

It is expected that by the year 2022, 1.5 billion devices will be connected to the internet. To deal with increased demand of mobile data from devices and users, 5G networks promise increased speeds and reduced latency to deliver overall improved mobile data services to devices and users. Sophisticated Technologies such as beamforming, massive MIMO, and dense small cells are among some that are critical to deliver the improved mobile data services. However, these technologies also increase the complexity of system architecture, which in turns complicate the maintenance of the system, especially dealing with a large scale network with intermittent hardware faults or ill configured network settings.

Currently whenever abnormal operations were detected or reported, a team of individuals considered domain experts in the field is called upon to investigate the issues and resolve the problem. These experts are highly trained and specialized to diagnose problems on a cellular network so that any issue can be quickly identified and rectified to reduce the downtime of the network. The process involves in reviewing significant amount of KPIs which is time consuming. Besides, it also takes an extraordinary amount of experience to learn the minute details of a network to be able to determine the exact cause of an error. Being able to automate the common fault detection for cellular networks will significantly reduce the expert involvement in maintaining the network operation. Prior papers have attempted the untapped field of automated fault troubleshooting. Traditionally, the automated fault troubleshooting in cellular networks is done based on logics in the form of rules or algorithms. The design of these logics are mainly based on expert knowledge. While this approach remains effective for some specific issues, it does not scale well with increasing complexity in the system and network. Applying machine learning to network management including fault management has recently emerged as an attractive approach to deal with well defined and perhaps also unknown faults [1]. Machine learning techniques applicable to fault management can be generally divided into two types. The first type uses analytical techniques where the system is trained using existing data and then perform analysis on live data to detect any fault. This technique either supervised or unsupervised learning is commonly used in detection and diagnosis [1]. The second type employs active techniques where the system is trained to take appropriate actions subject to a feedback and reward system. This technique such as reinforcement learning have been exclusively applied to automatic corrective actions to compensate for faults [1]. Our work falls under the first type.

In cellular networks, various key performance indicators (KPIs) are constantly reported by the system. When properly analyzed, these KPIs can indicate performance issue and hardware failure to some extent (see [1] and the references therein). Some of the earlier works were done largely based on statistical model, such as Barco *et al.* [2], [3] applied Naive Bayesian approach to achieve automated diagnosis in cellular networks, and Yang *et al.* in [4] applied big data analytics approach to perform anomaly detection and RCA for cellular networks. Recently, machine learning technique has been used to deal with the fault management in cellular networks. Gomez-Andrades *et al.* proposed using unsupervised learning technique to identify anomaly and subsequently be labelled by experts for future reference [5]. The method uses advanced statistical model to analyze the KPIs for model tuning. While encouraging outcomes were reported, the downside of the method is the need for expertise in cellular networks, statistics and machine learning domains to tune the system. In [6],

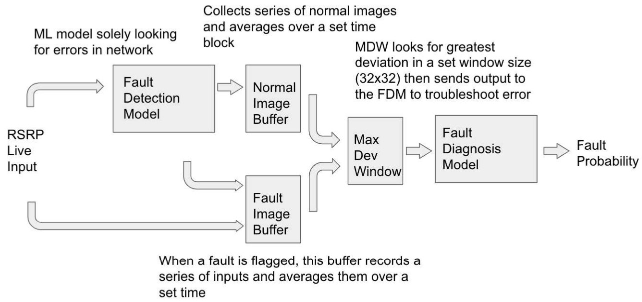


Fig. 1: Illustration of the system model.

we focus on training a High-Order Recurrent Neural Network to identify performance degradation based on a sequence of Reference Signal Received Power (RSRP) reported by users with promising outcome. These recent works have confirmed that machine learning technique can adequately deal with fault management in cellular networks to some extent. Unlike the traditional rule base approach where increase in complexity of cellular network design directly adds complexity to the rule base design, machine learning technique is less influenced by the underlying design of cellular networks, and hence a more future-proof solution for fault management in cellular networks.

In this paper, we use the RSRP reported by users to construct a heatmap for a specific region covering a set of base stations. Due to the noise in the RSRP readings, analyzing directly the snapshot of a heatmap to detect potential faults may produce excessive false positives due to noise. Our design considers two stages where the first stage deals with detection of a degradation event, and the second stage deals with fault identification. This paper focuses on the second stage where we design and train a neural network to classify some common faults of a base stations which can be reflected on the RSRP readings. In the next section, we describe our approach where the system model, the dataset and the preprocessing of the data are explained. Section III present and compare two machine learning models for the fault classification. Results are discussed in Section IV with important conclusion drawn in Section V.

II. Our Approach

A. System model

In our earlier work given in [6], we demonstrated using a sequence of reported RSRP for performance degradation to the identification of several common faults using a snapshot of RSRP gather within a region. The approach taken in this paper is to implement a twofold system consisting of two stages presented in Fig. 1 where we separate the process of identifying a performance degradation and analyzing the fault.

The first stage is a binary classifier responsible for differentiating between normal and faulty conditions in the network. Under the condition without no influence from a fault, the classifier indicates a normal network operation. During a normal operation, the system constantly collects the snapshots of RSRP images of the region to produce a heatmap image. Due to noise, fading, and hardware implementation variations on a user equipment (UE), the produced image of heatmaps changes from time to time. The system retains a number of last seen heatmap images in the buffer. These images can be used to produce an overall average heatmap image representing the normal operation at the time of the production.

A fault in cellular networks can be detected in many ways (see [6] for a detailed discussion). In [6], we propose using a High-Order Recurrent Neural Network to detect performance degradation, which will be used as a classifier in our system. In the classifier, a trained High-Order Recurrent Neural Network is implemented in each base station within the region to detect performance degradation. It constantly collects a series of RSRP readings from a set of UEs and reports whether a certain degree of performance degradation has happened. When a performance degradation exceeds a certain threshold, the classifier triggers a faulty condition and the system proceeds to the second stage. While the High-Order Recurrent Neural Network can detect a potential fault, it is not designed to identify the cause. An additional effort is required to evaluate the cause of the fault. This task is performed at the second stage of the system.

As a preparation for the second stage, the system immediately produces the overall average heatmap image for the normal condition and starts collecting heatmap images for the faulty condition. After collecting a sufficient number of heatmap images representing the faulty condition, an overall average heapmap image for the faulty condition is generated. The purpose of collecting a sufficient number of heatmap images is to reduce the impact of noise, fading and hardware implementation variations.

The two averaged heatmap images from the buffers will then be sent to an additional layer to process the regions of highest deviations. The maximum deviation window will find the areas in the heatmap where the greatest variance between the averaged normal and faulty heatmap images exists. The areas swept by the maximum deviation window that have the highest variance will be captured, and will create a new sample that can then be sent to the fault diagnosis model. In this stage, a machine learning model will analyse which fault the network is experiencing and output its result. In this paper, we focus on developing a machine learning model to identify faults in the heatmap. The models explored for the fault diagnosis layer include random forest and a convolutional neural network. The result of this twofold system is increased accuracy when examining errors in a network, and will eventually allow the system to find the locations of faults.

B. Dataset Collection and Preprocessing

In this paper, we use ATOLL network simulator [7] to produce the heatmap of a service area. The heatmap is generated by collecting the reference signal received power (RSRP) at various locations. Pixel values for the heat map correlate directly with power intensities experienced at locations on the service area. Images in this paper have been converted to grayscale for two reasons. Firstly, color in the heatmap has little significance as the colors are mapped from a one-dimensional power intensity level. Secondly, grayscale images reduce computational complexity. Each grayscale heatmap image is converted to a flatten one-dimensional representation that corresponds with location on the map before being fed into a machine learning model. Fig. 2 illustrate an example of the grayscale heatmap produced by ATOLL simulation. The service area under consideration is the city of Brussels covering over 800 km^2. The area consists of 97 sectorized base stations, each base station has three transmitters. In our experiments, we focus on altering the operation of 40 base stations or 120 transmitters, while keeping other base stations functioning as normal.

Each heatmap image represents a snapshot of the RSRP readings collected in the service area. These RSRP readings fluctuate over the time due to random noise, fading and different hardware architectures. RSRP readings fluctuation may cause a machine learning system to overfit on the training set. To reduce the impact of RSRP readings fluctuation influencing the performance of a machine learning system, a set of heatmap images collected over a period of time is used to produce an overall average heatmap image.

Table I shows the settings in simulation for the dataset collections. The following describes the faults considered in our experiments.

1) Site Outage: Each of the three transmitters on a base station are turned off.
2) Transmission Power: A power error is introduced to a transmitter varying the default value of 43 dBm to 25 dBm, 29 dBm, and 35 dBm.
3) Antenna Uptilt (AU): The tilt angle of the transmitter was increased from 0 degrees to positive 25 degrees.
4) Antenna Downtilt (AD): The tilt angle of the transmitter was decreased from 0 degrees to negative 25 degrees.

Upon detection of a fault, the system enters the second stage to perform fault classification. Instead of presenting the entire heatmap image to the machine learning model which consists of nearly 100 base stations, a comparison between a normal and a faulty image is performed. In this paper, we produce the difference between the normal heatmap image averaged over a period of time and the faulty heatmap image averaged over the same duration. The comparison allows removal of many regions where base stations operate normally. We design a maximum deviation window sweeping to identify the region that is likely to produce the fault. This information is further checked against the binary classifier reporting to confirm that the identified region indeed covers the potential faulty base station reported by the binary classifier. Any potential false alarm is removed from this stage.

The resulting image from the maximum deviation window is a 32 by 32 grayscale image showing the image difference between the average normal and average fault images. Fig. 3 shows an instance of site outage after captured by maximum deviation window.

III. Machine Learning System Setup

In this paper, we use Convolutional Neural Network (CNN) and Random Forest to perform classification of detected faults. As mentioned earlier, we train the machine learning system to classify four faults.

A. Convolutional Neural Network

CNN is a powerful machine learning system where layers of multiple nodes, called perceptrons, are connected to each perceptron in the next layer. Each connection between perceptrons has a corresponding weight which is multiplied by the output of a node before going into the next node. Before being multiplied by weights, outputs of perceptrons are sent through an activation function. The activation function adds non-linearities for the model to understand complex nonlinear data. The input of each perceptron is the sum of all perceptron outputs from the previous layer which has been multiplied by its corresponding weights and sent through their activation function. During training the model uses a technique called

TABLE I: Simulation settings.

System Parameters	Values
Cellular Layout	120 Macrocell sites
Sectors	3 sectors per BS
Simulation Area	800 km^2
Path Loss Model	Ray-Tracing
BS Transmit Power	43 dBm
Cell Individual Offset	0 dBm
Antenna Tilt	0 deg
Antenna Gain	18.3 dBi
Carrier Frequency	2100 MHz

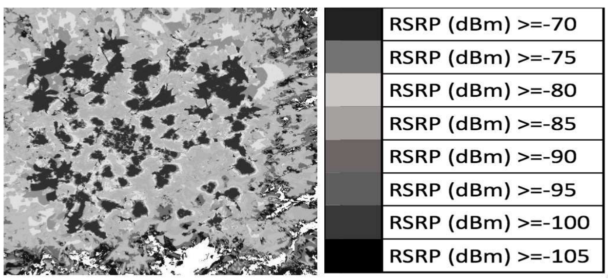


Fig. 2: Heatmap image produced by ATOLL simulator.

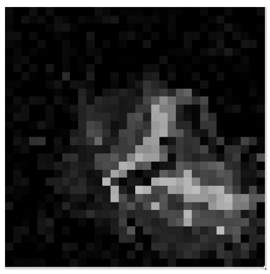
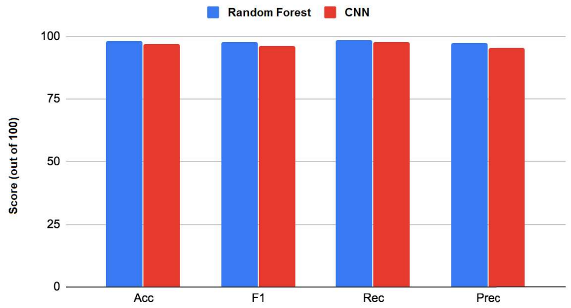


Fig. 3: Illustration of site outage deviation captured by the maximum deviation window.



tb]

Fig. 4: Model scores of CNN and Random Forest for four different faults.

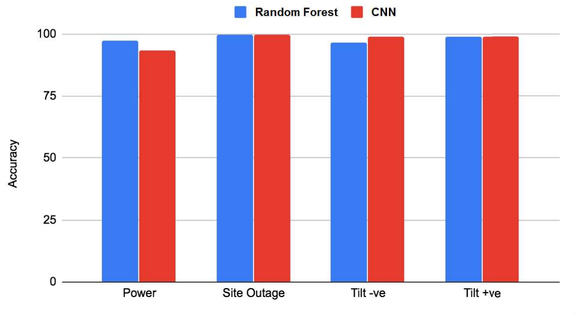


Fig. 5: Fault classification accuracy of CNN and Random Forest for four different faults.

backpropagation where the model learns to minimize errors by adjusting the weights of each connection and bias to each perceptron.

In this paper, a simple four layer network is designed. The input layer is the size of the images used, in this case, 32x32 or 1024. The subsequent number of perceptrons in each layer are 512, 128, 64, 32, and 4. The numbers are selected to remain in the performance region of the computer and are large enough to reduce the risk of bottlenecking the CNN. Other configurations are also tested with little to no additional improvement. The activation function chosen is LeakyReLu to preserve any negative input information into other layers, which give a slight improvement in the accuracy over the standard ReLU.

B. Random Forest

The other classifier used in this paper is a Random Forest. A random forest is an aggregate of decision trees where decision trees take in features and perform splits based off what is known as the CART algorithm in order to minimalize the Gini Impurity. Each split will occur until the decision tree produces the lowest error and separate classes in the purist manner. This is done very efficiently to the point until decision trees begin to overfit to the training set and reduce reliability of the model.

To counteract the overfitting characteristic of decision trees a random forest employes what is called an ensemble, where multiple trees are trained individually. In a process called bootstrapping, the trees will each receive a different subset of the dataset to be trained. This ensures each decision tree is unique and learning something different about the feature in the data. An additional measure to reduce the risk of overfitting is to prevent one tree from making a decision on the classification. Instead each tree casts a vote, and determining factor for which class is selected depends on the highest vote.

IV. Result Discussions

In our experiment, we configure ATOLL simulator to produce normal RSSI heatmap images under normal condition and the conditions containing either one of the four considered faults. These images are then used to train and test the accuracy of our developed machine learning system. The RSRP readings includes Gaussian noise which has astandard deviation set to 5dBm. We first compare the machine learning model scores in Fig. 4. It is shown that the random forest model outperforms CNN in almost each of the categories of measurement. Not only does the random forest have a higher accuracy when making predictions of the faults in the network but it is also shown that random forest has higher recall, precision, and F1 scores than CNN. A comparison of accuracy for different faults are given in Fig. 5.

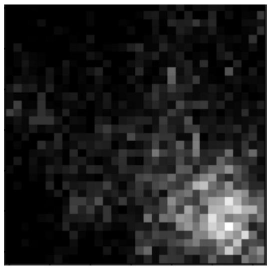


Fig. 6: RF feature importance heatmap.

Random forest offers better performance than DNN because the random forest can potentially filter out the noise better than CNN. As has been explained earlier in this paper, random forest takes in a set of features and makes splits based off the features in the dataset. Splits performed are based off creating if/than rules while the CNN is looking for optimum weights to reduce error of a pixel.

In the case of images each of the pixels correspond to a pixel sent in the sample. Naturally this means the total feature size is the number of pixels in a given window. Random Forest has an ability to determine which features are the most important when making decisions where to split and categorize classes. The number of times a feature is used by decision trees in a Random forest the more important the feature. Fig. 6 is a reconstructed heatmap of each of the features the random forest sees as important when making decisions. The brighter the pixel in the window indicates that the random forest relies more on those pixels to make splits and build its trees. Although the image appears noisy, it is easy to see key

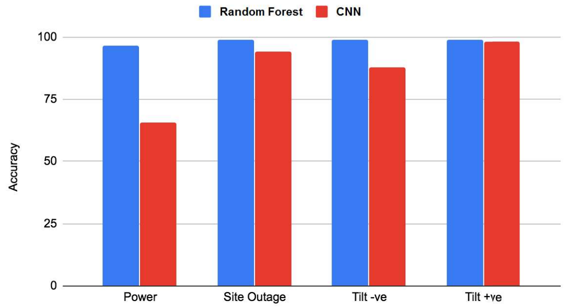


Fig. 7: Fault classification accuracy of CNN and Random Forest for four different faults (RSRP readings with Gaussian noise standard deviation = 10dBm).

areas the random forest uses to create its higher accuracy.

In Fig. 7, we increase the standard deviation of Gaussian noise on RSRP readings from 5dBm to 10dBm. We can immediately see the high impact of noise on CNN performance. The accuracy of CNN drops significantly particularly for the case of transmission power fault. As CNN is attempting to generalize over the whole image rather than seeking to find areas of interest, CNN find slight difficulty to cope with noise and hence its accuracy is not as good as random forest.

V. Conclusions and Future Work

Self healing will be a necessary component to reduce cost and time to solve the problem of unacceptable drops in service in a network. In this paper, we presented an automated solution to diagnosing faulty conditions leveraging the capabilities of machine learning. The twofold model approach brings promising accuracy and has future implications for identifying exact locations of network quality drop. Utilizing RSRP heatmap images, operators can consistently grab power levels transmitted and received by users in a service area to produce these maps, and when drops in service are detected, the operators are able to determine exact areas automatically.

As directly using instantaneous RSRP readings trigger high false alarm in fault detection and diagnosis, our proposed system is based on a twofold model which firstly detect potential performance issue and collect sufficient RSRP readings over a period to reduce the impact of noise, and secondly uses machine learning models to classify the fault for automatic fault diagnosis.

To build the system, we reused our earlier work on performance degradation detection for the purpose of fault detection. Once an event of performance degradation was detected, fault diagnosis process was triggered. We tested two machine learning models for the fault classification, including convolutional neural network and random forest. Our test showed that random forest was able to cope with noise presence in RSRP readings compared with CNN. When presenting noisy images, CNN struggled to identify area of interest and thus failed to accurately classify faults. In the future, we shall continue to enrich the types of faults that a machine learning system can classify.

References

[1] D. Mulvey, C. H. Foh, M. A. Imran, and R. Tafazolli, "Cell fault management using machine learning techniques," *IEEE Access*, vol. 7, pp. 124 514–124 539, 2019.

[2] R. Barco, L. Nielsen, R. Guerrero, G. Hylander, and S. Patel, "Automated troubleshooting of a mobile communication network using bayesian networks," *4th International Workshop on Mobile and Wireless Communications Network*, pp. 606–610, 2002.

[3] R. Barco, V. Wille, and L. Díez, "System for automated diagnosis in cellular networks based on performance indicators," *European Transactions on Telecommunications*, vol. 16, no. 5, pp. 399–409, 2005.

[4] Y. S. J. Y. Kai Yang, Ruilin Liu and X. Chen, "Deep Network Analyzer (DNA): A big data analytics platform for cellular networks," *IEEE Internet of Things Journal*, Dec. 2017.

[5] A. Gómez-Andrades, R. Barco, and I. Serrano, "A method of assessment of LTE coverage holes," *EURASIP Journal on Wireless Communications and Networking*, vol. 2016, no. 1, p. 1, Oct. 2016. [Online]. Available: http://dx.doi.org/10.1186/s13638-016-0733-y

[6] D. Mulvey, C. H. Foh, M. A. Imran, and R. Tafazolli, "Cell coverage degradation detection using deep learning techniques," in *Proceedings of the 9th International Conference on ICT Convergence (ICTC 2018)*, 2018.

[7] "Atoll RF design tool," https://www.forsk.com.