Journal of Computers Vol. 30 No. 5, 2019, pp. 268-278
doi:10.3966/199115992019103005021

A Small Cell Outage Prediction Method Based on RNN Model

Ming-Yen Wu*, Yu-Hsiang Lin, Tse-Hsiang Tseng, Chen-Min Hsu

Network Management Laboratory, Chunghwa Telecom Laboratories, Taoyuan 326, Taiwan
{marcuswu, taddylin, tsehsiang, paulhsu}@cht.com.tw

Received 15 March 2019; Revised 30 June 2019; Accepted 4 September 2019

Abstract. Small cells play an important bridging role in the nearly 5G network to connect the various IoT devices and internet of vehicles for low-latency and high-speed communication. The small cell outage diagnosis based on RNN model is proposed to improve the wireless network quality and customer experience due to the most small cells are deployed in the high-density population area. This research is used to diagnose the small cell obstacle events based on KPIs information and deep learning algorithm. The device KPIs are used as input factor in RNN deep learning model and the experiment result show the accuracy of validation dataset is to be 96%. The prediction result can repair the outage and adjust the parameters in advance according to the correlation between KPIs values change and outage events.

Keywords: deep learning, outage diagnosis, RNN, small cell

1 Introduction

The 5G new radio (NR) standards are released in the 3GPP Release-15 in June 2018 [1] that means the 5G era is going to implement in our life. The 5G standalone (SA) reduces the reliance on 4G network and has better performance in terms of large bandwidth, low latency, and high-speed communication. In addition, the 5G applications will connect more devices to realize IoT, internet of vehicles (IoV), and smart city services. In the early days of 5G network construction, it is necessary for small cells that enhance the service coverage, and provide a stable wireless network environment. The low-cost and rapid deployment are characteristic of the small cells that quickly provide the wireless services from a few meters to a few kilometers, and suitable for building in high-density population areas [2].

When the outage of the small cell had encountered, the telecom operator needs to find out the obstacles immediately to ensure the services for functioning properly in view of the fact that the small cells are mass deployed around the world [3]. The root obstacles have complex factors such as the LTE core network, eNodeB gateway, network devices, or communication outage between each facility. Therefore, finding the cause of small cell outages in the shortest time will effectively help the telecom operators improve the user experience.

Additionally, data routing and switching are much more complicated with the evolution of wireless network architecture. There are more costly and time-consuming in the diagnostic device node based on traditional rule-based solution for small cell outages. Therefore, the emerging deep learning methodology helps telecom operators build knowledge-based diagnostics on 5G networks.

This paper structured as follow. Section 1 introduces the research background and motivation. Section 2 describes the related works of deep learning methodology and self-organizing network. Section 3 describes the system design, architecture, and research method. Section 4 describes the experimental results. Section 5 proposes conclusions. Finally, Section 6 proposes future research.

* Corresponding Author

2 Related Works

The 3GPP organization had defined the self-organizing network (SON) whose spirit was to achieve effective wireless transmission efficiency and improve stability under limited spectrum resources [4]. The purpose of SON was transforming the initial setting, configuration, repairment, and optimization into automated processes for reducing the network management and labor costs of telecom operators.

The Q-learning technology, one of deep learning branch, has been adopted in some SON research to manage the wireless access networks. Therefore, there are some deep learning applications have been used in the wireless network to improve the quality of services.

2.1 Self-Organizing Network (SON) in Malfunction Detection

The SON has three major functionality modules such as self-configuration, self-optimization, and self-healing. The self-healing focuses on automatic detection, cause diagnosis, and troubleshooting of base station.

Bandera had proposed a cell outage detection algorithm based on incoming handover statistics in 2005 [5], and the neighboring measurement method was used to detect outage in two cases in this algorithm. First, when the cell in outage was able to report performance indicators; second, when these indicators were not available because the base station was affected. The total number of current and previous handovers of all neighboring base stations was calculated in this algorithm, and it judged whether there was the malfunction in the neighboring base stations based on the numerical change in statistical results. In addition, it is judged by the LTE key performance indicators (KPIs) whether the service interruption of the base station is directly controlled by the telecom operator or equipment manufacturer. However, there needed to set up a monitoring period value to determine the algorithm execution period in this algorithm. The algorithm was executed frequently when setting the lower monitoring period value, but it also increases the effort of the operations support system (OSS). The default monitoring period value was one hour in this paper but still depended on the frequency of updating KPIs. In the research, if the KPIs information could not be obtained, it was judged to be the malfunction of the base station. However, this judgment will make mistakes when the service is still working normally but the KPIs just missing updates.

Ciocarlie et al. proposed a detecting anomalies in cellular networks using an ensemble method in 2013 [6]. This method aggregated some analytical models and applied in the following three cases. First, when analyzing the KPIs in univariate time series; second, when analyzing the KPIs in multivariate time series; third, assembled two cases with configuration management data, performance management data, and fault management data. The empirical distribution function and Kolmogorov-Smirnov test had been used in this paper, and those were grouping the KPIs of base stations and find the center of each cluster. In addition, the support vector machine (SVM) was used to divide the KPIs cluster centers into two categories, one was a normal category and the other was an abnormal category. Finally, the autoregressive moving average (ARMA) method was used as a prediction model to predict whether KPIs information appeared in the cluster center characteristics of the abnormal class. However, this algorithm could not be executed if the KPIs information could not be obtained from the base station.

In addition, Keshavamurthy and Ashraf proposed a conceptual design of proactive SONs based on the big data framework for 5G cellular network in 2016 [7]. The framework was based on big data analysis and machine learning technology which presented that the end to end devices will various connects in 5G network. With more complex application scenarios cause the device connection also more complicated. The traditional SON framework adopted the passive maintenance strategy to repair and optimize the malfunction after it was detected. But the traditional framework had been unable to handle the massive number of devices, large bandwidth, and low latency wireless network architecture. Therefore, the research had presented the new framework concept based on Game-Playing AI engine and divided into three items. First, the data reduction strategies; second, the heuristic analysis based prediction; third, the statistical root cause analysis. The data reduction strategies mean that the OSS will access and store the hug KPIs information. Therefore, the bit-map scaling algorithm and windowing technology were used in the research to store the KPIs information, and the search performance and access space could be optimized. The traditional regression model was used as a prediction model such as linear regression and logistic regression to construct the features extraction method and machine learning model. However, the traditional regression method was easily overfitting when predicting the data tendency in this paper.

Gandalf project was a massive OSS to monitor the wireless network that was proposed by Altman et al. [8]. This project was used to manage the wireless resource and evaluate the quality of service (QoS), and the QoS evaluation was implemented by traditional Bayesian network architecture. However, the traditional Bayesian network was most suitable for the case where each feature was discrete, it was not necessarily suitable for the non-discrete of the feature such as KPIs information.

Most of the above research adopted the general statistical methods such as regression model and Bayesian network to detect the malfunction of the cellular network. Recently, the SON related research had appeared machine learning algorithms such as Q-Learning and reinforcement learning.

3 A Small Cell Outage Perdition Based on RNN Model

This research was based on the deep neural network (DNN) and long short term memory (LSTM) applied to self-healing to strengthen the network diagnostic capability of small cell.

3.1 The Network Architecture and Outage Events of Small Cell

The small cells are connected to the security gateway (SecGW) via WAN and then connected to the eNodeB (eNB) Gateway via the SecGW, and finally extended to the LTE core network of evolved packet core system (EPC), as shown in Fig. 1. The element management system (EMS) managed all small cells via TR-069 protocols [9] and provided the network management capability such as provisioning, monitoring, parameter configuration, and performance management. The small cells will upload the performance management (PM) data and configuration management (CM) data to EMS every minute. The EMS calculated the KPIs of each small cell based on the PM and CM data, and the LTE KPIs specification which was defined in 3GPP [10-11]. The centralized SON (C-SON) system connected the EMS via API, and which accessed the PM, CM, and KPIs information to tune the parameters and optimize the small cell performance.

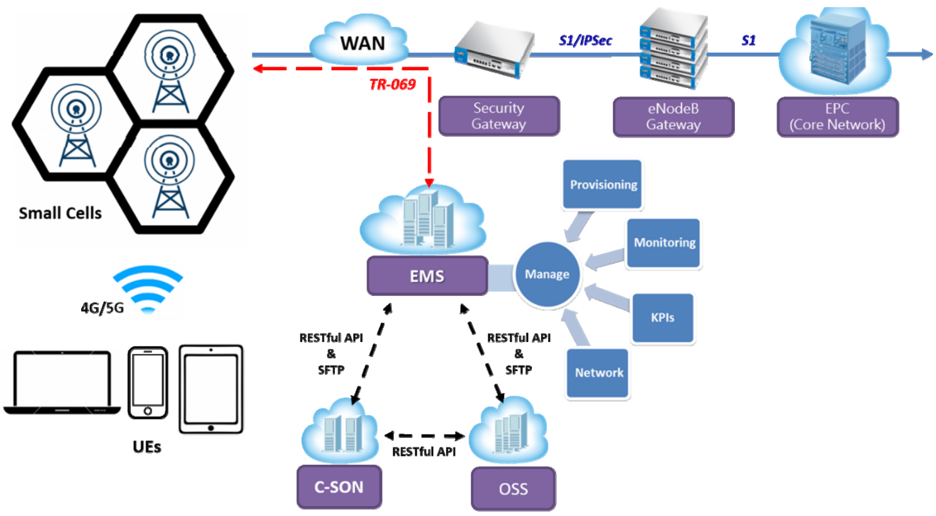

Fig. 1. The small cell network architecture

The small cell device contains the following 11 outage events: PCI conflict, IP address conflict, MAC address conflict, CPU overload, memory overload, PCI unavailable, time sync failure, over-high temperature, firmware upgrade failure, X2 interface link disconnection, and PM/CM uploading failure. All of the above outage events were event logs generated by the small cell. Those malfunctions will seriously affect the quality of wireless network services and even unable to provide the service.

There were contains the following 4 outage events when EMS managed all small cells: provision failure, parameter getting failure, parameter setting failure, and PWS parameter setting failure. All of the above outage events were event logs generated by the small cell. Although the malfunctions did not affect the small cell to provide the wireless network services but could not monitor and optimize the small cell performance in real time.

There were contains the following 3 outage events when the small cell connects to the SecGW: IPSec tunnel establishment failure, IPSec tunnel unexpected disconnection, and S1-MME interface link disconnection. All of the above outage events were event logs generated by the small cell. Those malfunctions will seriously affect the quality of wireless network services and even unable to provide the service.

The related equipment and system event logs and fault management information were being collected in this research. There were a total of 18 kinds of outage events were summarized, and recorded the event timestamp, event source, and the corresponding device ID of small cell. The total malfunctions were shown in Fig. 2.

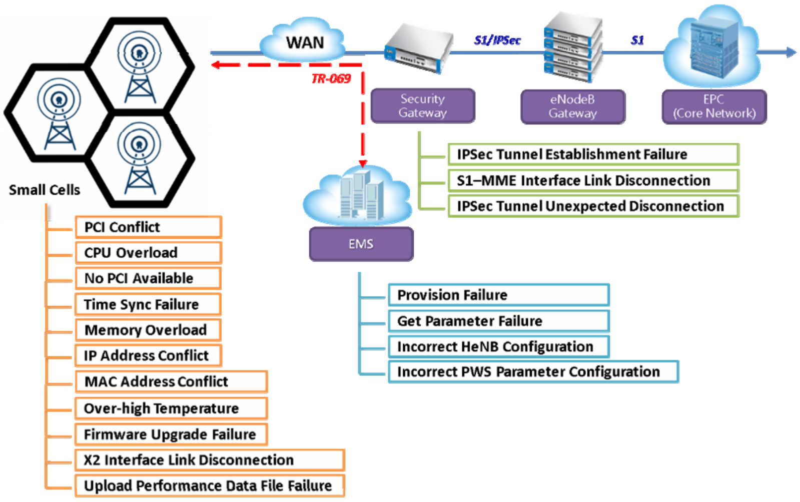

Fig. 2. The malfunctions in small cell

3.2 The System Architecture

The small cells periodically uploaded the PM and CM data to auto-configuration server (ACS) ever minute via the TR-069 protocol and stored the data in the repository, and the outage events also forwarded to ACS and store in the ACS database. The ACS received around 200+ counters of PM and CM, transferring all counters to diagnosis server (DS) via RESTful API. As the same time, The PM and CM counters were converted into the LTE KPIs via KPIs formula database, presenting the wireless network quality information of all small cells. The DS was based on the ELK architecture (Elasticsearch, Logstash, Kibana), an open source centralized logging management, to store 89 KPIs and outage events of all small cells, and executing the data pre-processing, mergence, model building, and classification prediction for outage diagnosis. The system architecture was shown in Fig. 3.

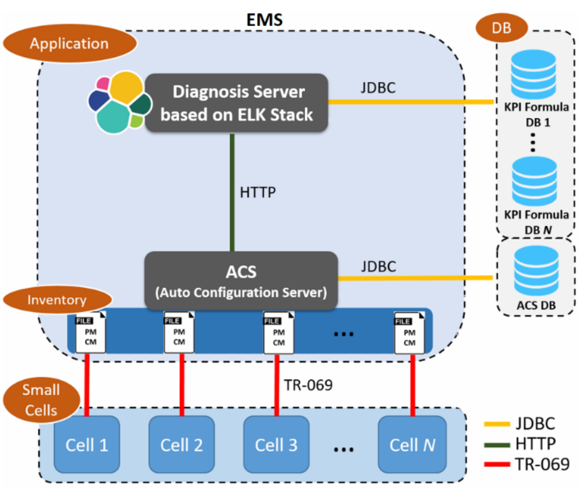

Fig. 3. System architecture

3.3 The Data Processing of KPIs and Outage Events

The wireless network quality information of the device was represented by 89 KPIs and contained the timestamp and device ID of all small cells, as shown in Table 1. Each LTE KPIs had its corresponding formula, and the parameters in the formula were the PM and CM counters that the device uploading.

Table 1. The LTE KPIs item and formula sample

KPIs Item / Timestamp	Formula / Format	Unit
Cell Availability	(RRU.CellAvailableTime/SampleTime)*100%	%
Ratio of Successful RRC Establishment	(RRC.SuccConnEstab/RRC.AttConnEstab)*100%	%
LTE HO Success Rate	(HO.FromeNBSucc.sum/(HO.FromeNBSucc.sum+HO.FromeNBFail.sum))*100%	%
Ratio of RRC Drop	(CONTEXT.AttRelEnb.Abnormal/(CONTEXT.AttRelEnb+RRC.ConnCurrent))*100%	%
RRC Normal Release Rate	((CONTEXT.AttRelEnb-CONTEXT.AttRelEnb.Abnormal)/ (CONTEXT.AttRelEnb+RRC.ConnCurrent))*100%	%
UL TTI Usage	(RRU.UsedPrbUL/RRU.AvailPrbUL)*100%	%
DL TTI Usage	(RRU.UsedPrbDL/RRU.AvailPrbDL)*100%	%
Date Time	YYYY-MM-DD HH: MM: SS	N/A

The malfunctions were divided into 18 categories (see Sect 3.1) and the example event logs were shown in Table 2. Each outage event log was recorded in the ACS database then DS extracted the newest log from the database. The logs were recorded the outage description, timestamp, and device ID of small cell. The KPIs and outage events could be merged with timestamp and device ID.

Table 2. The event log sample

	Outage Event Description
(1)	Error Exception: 2018-05-29 14:28:08.000 the device A43142-LTE had occurred CPU overload from ip 192.168.3.49……
(2)	Error Exception: 2018-05-29 14:28:34.000 the device A43142-LTE had occurred memory overload from ip 192.168.3.49……
(3)	Error Exception: 2018-05-29 14:30:48.000 the device A43142-LTE had occurred over-high temperature from ip 192.168.3.49……
(4)	Error Exception: 2018-06-14 09:05:18.000 the device B23441-LTE had occurred S1 interface link disconnection from ip 192.168.3.251……
(5)	Error Exception: 2018-07-05 23:44:17.000 the device B22539-LTE had occurred gateway ipsec tunnel disconnection from ip 192.168.3.149……
(6)	Error Exception: 2018-07-18 11:08:10.000 the device A37428-LTE had occurred (HO) X2 interface link disconnection from ip 192.168.3.121……

In addition, the mergence processing will have 2 identical KPIs corresponding to 2 outage events when there was the same device had occurred 2 outages in 1 minute. The above scenario liked (1) & (2) in Table 2 that the small cell A43142 had 2 outages in 1 minute, one was CPU overload and the other was memory overload.

Clean data. The PM and CM counters caused the denominator to be 0 when DS calculated the KPIs in sometimes. So, the average value of KPIs for each category was adopted to fill up the missing value of the above scenario in this research. In addition, the principal component analysis (PCA) method was adopted to filter the outlier KPIs values or noise values.

Scale data. In this research, some KPIs values were large such as the interface uplink and downlink throughput (Kbits); some KPIs values just used 1 or 0 to represent the true or false. So, the normalized skill was adopted to normalize all KPIs values to [0, 1], and accelerated the model convergence.

Transform data. The malfunction logs were divided into 18 categories and 1 non-malfunction category. Therefore, there were a total of 19 categories in the research. The outage event logs were text and needed to be parsed as the word vectors as the supervised labels, and merged the outage events with KPIs by timestamp and device ID.

Label encode and One-Hot encode. The label encoding skill was adopted to use digital number represented the label name, and also adopted the One-Hot encoding skill to convert the categories to the N*19 dimensions.

3.4 Training Model Building

This research found that a single outage event could be associated with multiple KPIs values were changed at the same time. For example, if a small cell was connected to the SecGW and suddenly occurred the S1-MME interface link disconnection, caused the KPI – Bytes Received of S1 Link Interface had decreased, and the KPI – Cell Availability had also decreased, as shown in Fig. 4. However, the user equipment (UE) was still connected on this small cell, so the KPI – Total UE Connected was never changed. Therefore, that was known from the above scenario that the KPIs values changed were associated with an outage event. In addition, there was also a time correlation between outages. For example, there were 2 outages from the same device in 1 minute such as (1) and (2) in Table 2. The outage (1) was CPU overload and the other outage (2) was memory overload. Those 2 outages had the causal relationship on the timestamp, as shown in Fig. 5.

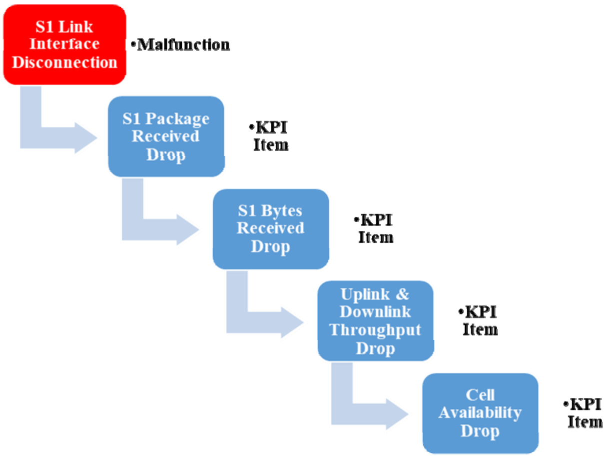

Fig. 4. Multiple KPIs drop by single malfunction

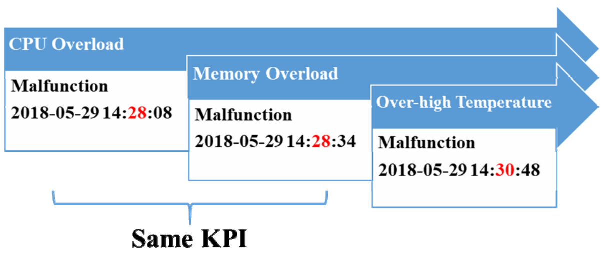

Fig. 5. Malfunction correlation

Additionally, most outages had occurred at the regular rush hour, as shown in Fig. 6. The most connection outages such as S1-MME Interface Link Disconnected and IPSec Tunnel Unexpected Disconnected had occurred disconnection at high KPI values - RRC connected users. Therefore, the long term short memory (LSTM) skill, a recurrent neural network (RNN) transformer, was adopted as the model base for KPIs information as time series data.

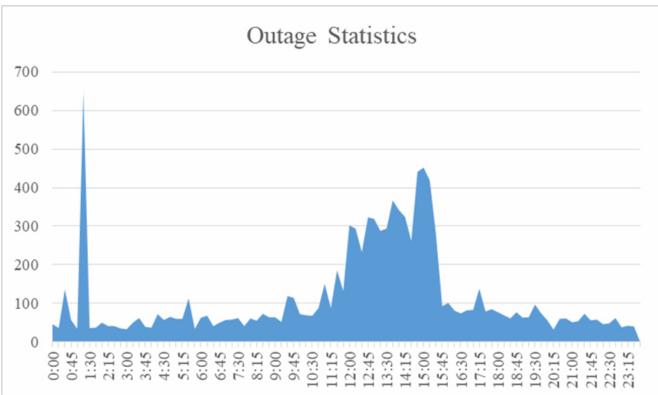

Fig. 6. Outage statistics for each hour

The dataset after processing was divided into training data and validation data. The training data was trained by 2 LSTM layers and added 4 hidden layers. Finally, the 18 outages categories and 1 non-outage category were classified by the Softmax algorithm. The validation data was used to verify model accuracy, precision, and recall based on multiple iterative training, and the new KPIs as testing data to verify the final model performance after training. Each hidden layer was added the dropout layer to prevent model overfitting. Finally, the class weight had been adjusted in this paper to improve the model precision for the imbalanced dataset. The training model architecture in this paper was shown in Fig. 7.

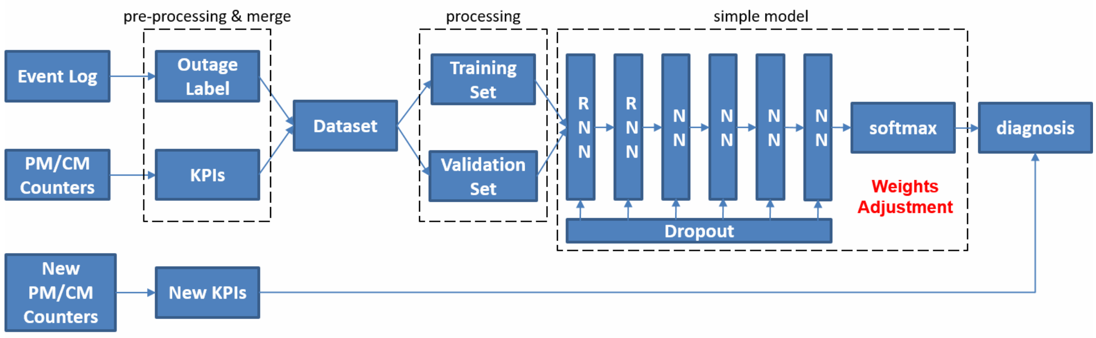

Fig. 7. Training model architecture

3.5 Category Weight Adjustment Method for Imbalanced Dataset

In this case, there were a total of 19 categories which the non-outage category was the majority in imbalance dataset. The category weight adjustment method was proposed in this section.

In Keras framework, there were the *class_weight* and *sample_weight* that two parameters could be assigned the weights of each category when fitting the model. We proposed an algorithm which designed each category weight as following pseudo-code.

```
var class_sampleSize = a dictionary (or map) which stores each class
label and corresponding sample size.
var total_sample = a total sample size.
var class_label = an array which stores all class label.
set class_weight = dict();
```

set *mu* = $\dfrac{total\_sample}{\max\{class\_smapleSize.values\}}$

```
for label in class_label:
```

    class_weight[label] = $\log_{m_u} \dfrac{total\_sample}{class\_smaple}$

```
set model.fit(…, class_weight=class_weight)
```

First, we designed a dictionary called *class_sampleSize* which stored each class label and corresponding sample size of training dataset (e.g., {c0: 2813, c1: 78, c2: 1014, c3: 510, c4: 7914, c5: 348}). The *total_sample* was the total sample size of training dataset (e.g., 12677). The *class_label* was an array which stored all class label name (e.g., [c0, c1, c2, c3, c4, c5]). The *mu* was the ratio of the largest category in *class_sampleSize* (e.g., 12677/7914 = 1.6018). Finally, each category weight was designed that the ratio of each category in *class_sampleSize* and used the logarithm to normalize each weight (e.g., the c0 weight $= \log_{1.6018} \dfrac{12677}{2813} \approx 3.1956$).

In some open source project, the automatic initial *class_weight* adopted the average size of category as the denominator to calculate each category weight, or there were no weights in default. For example, the initial $mu = \dfrac{12677}{6} \approx 2112.83$, and the weight of largest category c4 was $\dfrac{211283}{7914} \cong 0.27$, and the weight of least category c1 was $\dfrac{211283}{78} \cong 27.09$. In scikit-learn, the default weight of each category was 1. So, we used $\log_{mu}$ to minimize the weight of the largest category as 1, and also normalize the scaling for each weight. In this sample, the weight of the largest category c4 was $\log_{1.6018}\frac{12677}{7914}=1$, and the weight of least category c1 was $\log_{1.6018}\frac{12677}{78}\approx 10.81$. The weight of each category will not be scaling too much to avoid training data deviation. Therefore, the smaller ratio category had a bigger weight.

4 Experiment Environments and Results

This research was based on the online small cell service and collected the relevant PM data, CM data, FM data, and outage event logs. The experiment results were based on the research method in Sect 3.

4.1 Experiment Environments

The 800 small cells had been deployed in Taiwan public areas such as supermarket, banquet, Taiwan high-speed rail, hospital, and police station. The device PM data, CM data, FM data, and malfunction logs were collected from 2018/5/1~2018/8/31 for a total of 123 days. There was 5,050,010 KPIs dataset after data processing and data mergence, and each KPI contained the 89 KPI items as the input features of the training model. Among that KPI dataset, there was 10,986 KPIs had the corresponding outage events. There were a total of 19 categories which contains 18 outage categories and 1 non-outage category (see Sect. 3.1 & 3.3).

In addition, the Tensorflow and Keras as the deep learning framework were adopted in this research and equipped with Nvidia GeForce graphics card to improve the computing performance.

4.2 Experiment Results

This experiment had used the previous period KPIs to predict next period (or next minute) status of small cell service, and the experiment had 2 test cases: first, the class weights had no adjusted; second, the class weights had adjusted.

The training dataset, validation dataset, and testing dataset were divided as the following:

- Training Dataset: 5/1~8/15. There were a total of 4,099,340 KPIs and 9,834 corresponding outage events.
- Validation Dataset: 8/16~8/30. There were a total of 891,815 KPIs and 1,053 corresponding outage events.
- Testing Dataset: 8/31. There were a total of 58,855 KPIs and 99 corresponding outage events.

In addition, the training model (see Sect. 3.4) configuration as the following:

- Epoch: 1,000~1,200
- Batch Size: 128~256
- Neurons: 64~128
- Activation Function: ReLU & Softmax
- Loss Function: Categorical Cross-Entropy

The experiment results were shown in Table 3. The validation accuracy with weights non-adjustment was 98%, and the precision was 27.5%, and the recall was 98%, as shown in Fig. 8 to Fig. 10. The non-outage KPIs were the majority in the training dataset. So, the training dataset was the imbalance dataset that caused the high-accuracy and low-precision in this case. Therefore, the weights adjustment was adopted to improve the above scenario.

Table 3. Experiment results

Test Case	Accuracy (val)	Precision (val)	Recall (val)	F1-Score
1. Weights Non-adjustment	98.3%	27.5%	98%	42.95
2. Weights Adjustment	96% (-2.3%)	53.5% (+26%)	92.2% (-5.8%)	67.66 (+24.71)

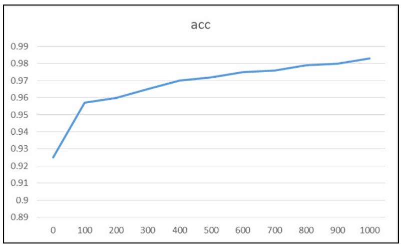

Fig. 8. Validation accuracy with weights non-adjustment

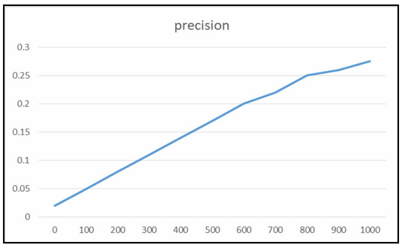

Fig. 9. Validation Precision with weights non-adjustment

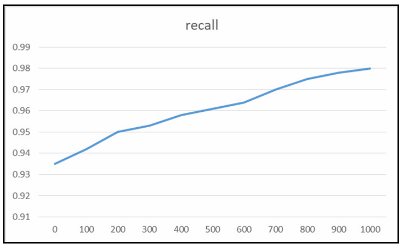

Fig. 10. Validation recall with weights non-adjustment

The validation accuracy with weights adjustment was 96%, and the precision was 53.5%, and the recall was 92.2%, as shown in Fig. 11 to Fig. 13. The threshold in the class weight matrix was raised for the non-outage category to prevent the low-precision. In the result, the accuracy and recall were decreased by 2% to 5.8% but the precision was increased by 26% that caused the F1-Score was increased by 24.71 points. Therefore, the weights adjustment was useful in this research and improved the prediction model performance.

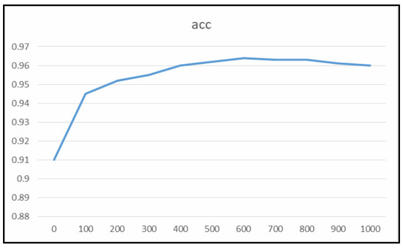

Fig. 11. Validation accuracy with weights adjustment

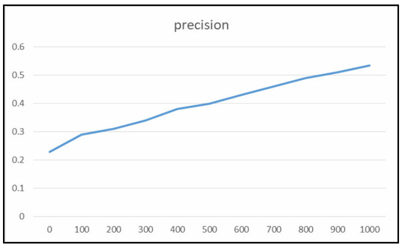

Fig. 12. Validation precision with weights adjustment

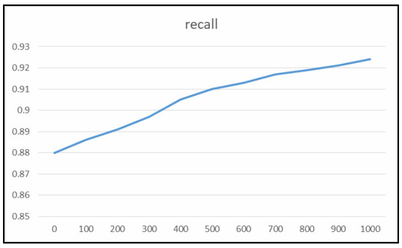

Fig. 13. Validation recall with weights adjustment

The confusion matrix of outage events for testing dataset was shown in Fig. 14. There were 99 outage events from 6 outage category as the following: IPSec tunnel establishment failure, IPSec tunnel unexpected disconnection, provision failure, S1-MME disconnection, time sync failure, and PM/CM uploading failure. The most outages belong to network disconnection such as IPSec tunnel disconnection and S1-MME disconnection. In confusion matrix, the testing accuracy was $\frac{89}{99} \approx 90\%$.

	O1	O2	O3	O4	O5	O6
O1	2	0	0	0	0	0
O2	3	33	0	0	0	0
O3	0	0	13	0	0	1
O4	0	3	0	29	0	0
O5	1	0	1	0	6	0
O6	0	0	1	0	0	6

O1：ipsec tunnel establishment failure
O2：ipsec tunnel unexpected disconnected
O3：provision failure
O4：s1 mme disconnected
O5：time sync failure
O6：upload pm data failure

Fig. 14. Confusion matrix of outage events for testing dataset

5 Conclusions

The small cell outage prediction method based on RNN model was proposed to maintain the quality of the wireless network and improve the customer experience. The RNN method could find out the relation of time-dependent features. The device KPIs and outage event logs of the online small cell were collected via data processing and deep learning stage. A category weight adjustment method was proposed to improve the model accuracy, precision, recall, and F1-score on the imbalanced dataset. In the experiment result, the validation accuracy prediction model was 96%, and the validation precision was 53.5%, and the validation recall was 92.2%, and the F1-score was 67.66 points. This research predicted the small cell service status with the next period (or next minute) via previous period KPIs values changed. Therefore, this research could adjust the configuration setting of the small cell before malfunction occurred, such as reducing the maximum number of UE connections.

This paper used machine learning techniques to predict the device outages in advance compared to other self-healing researches. In addition, this research still collected more device data or equipment configuration, and optimized the training model parameters. Moreover, reinforcement learning and deep Q-learning (DQL) is adopting in the new research to improve the model performances.

6 Future Works

This paper summarized the following works:

- Other outage events could be collected. There is more accuracy of outage classification when the clearer the outage event definition.
- The gateway data could be collected. There is the SecGW and eNB gateway in the network architecture which gateway data have more network information to add the features in the training model.
- The UE data could be collected. There is the UE information that stores connection data from UE in small cell CM data.
- The training model parameters would be tuned.
- The more outage data would be collected to improve the imbalanced dataset.

References

[1] 3GPP Technical Specification Group (TSG) Services and System Aspects (SA); standalone Release 15, 5G NR specifications, 2018.

[2] T. Nakamura, S. Nagata, A. Benjebbour, Y. Kishiyama, T. Hai, S. Xiaodong, Y. Ning, L. Nan, Trends in small cell enhancements in LTE advanced, IEEE Communications Magazine 51(2)(2013) 98-105.

[3] HUAWEI White Paper, Five Trends to Small Cell 2020, in: Proc. Mobile World Congress (MWC), 2016.

[4] 3GPP Technical Specification Group (TSG) Services and System Aspects (SA); Telecommunication management; Self-Organizing Networks (SON); Concepts and requirements, 3GPP TS 32.500 v15.0.0 Release 15, 2018.

[5] I. de-la-Bandera, R. Barco, P. Muñoz, I. Serrano, Cell Outage Detection Based on Handover Statistics, IEEE Communications Letters 19(7)(2015) 1189-1192.

[6] G.F. Ciocarlie, U. Lindqvist, S. Novaczki, H. Sanneck, Detecting anomalies in cellular networks using an ensemble method, in: Proc. 9th International Conference on Network and Service Management, 2013.

[7] B. Keshavamurthy, M. Ashraf, Conceptual design of proactive SONs based on the big data framework for 5G cellular networks: a novel machine learning perspective facilitating a shift in the SON paradigm, in: Proc. International Conference System Modeling & Advancement in Research Trends, 2016.

[8] Z. Altman, R. Skehill, R. Barco, L. Moltsen, R. Brennan, A. Samhat, R. Khanafer, H. Dubreil, M. Barry, B. Solana, The Celtic Gandalf framework, in: Proc. 2006 IEEE Mediterranean Electrotechnical Conference, 2006.

[9] TR-069 Amendment 5, TR-069 CPE WAN Management Protocol, Broadband Home Technical Working Group, Broadband Forum, 2013.

[10] Technical Specification Group Services and System Aspects; Telecommunication management; Key Performance Indicators (KPI) for Evolved Universal Terrestrial Radio Access Network (E-UTRAN): Definitions, Release 14, 3GPP TS

32.450 v14.0.0, 2017.

[11] Technical Specification Group Services and System Aspects; Telecommunication management; Key Performance Indicators (KPI) for Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Requirements, Release 14, 3GPP TS 32.451 v14.0.0, 2017.