Self-healing solutions for Wi-Fi networks to provide seamless handover

Lina Hao, Bryan Ng
School of Engineering and Computer Science
Victoria University of Wellington
Email:{lina.hao, bryan.ng}@ecs.vuw.ac.nz

Abstract—**The dynamic nature of the wireless channel poses a challenge to services requiring seamless and uniform network quality of service (QoS). Self-healing, a promising approach under the self-organizing networks (SON) paradigm, and has been shown to deal with unexpected network faults in cellular networks. In this paper, we use simple machine learning (ML) algorithms inspired by SON developments in cellular networks. Evaluation results show that the proposed approach identifies the faulty APs. Our proposed approach improves throughput by 63.6% and reduces packet loss rate by 16.6% compared with standard 802.11.**

I. Introduction

Self-Organizing Networks (SON) is a promising automation technology used to simplify planning, configuration, optimization and healing. The 3rd Generation Partnership Project (3GPP) has included SON as part of recent standards of mobile communications networks such as Long Term Evolution(LTE). The SON paradigm has outlines a set of principles and concepts to add automation to cellular networks requiring lower maintenance than traditional networks while improving QoS [1].

In this paper, we use machine learning algorithms to achieve self-healing in Wi-Fi networks. Several simple algorithms such as K-nearest neighbor (KNN), Support Vector Machine (SVM) and Local Outlier Factor (LOF) are used to detect and locate faulty APs based on the AP energy information in a typical campus Wi-Fi network. Upon detection of a faulty AP, a central controller retrieves the neighbor APs information from neighbor list to trigger handover for network compensation.

II. Background and Related Work

Algorithmic solutions to QoS in Wi-Fi have appeared in the literature at least a decade ago [11], [12]. The increasing popularity of machine learning techniques have seen researchers applying them for optimizing self-healing processes over wireless networks. Xue et al. [2] built a cell outage detection (COD) method based on the K Nearest Neighbor (KNN) algorithm while Zhang et al.[5] used another approach called improved local outlier factor (LOF) algorithm (M-LOF) to detect the cell outage based on the number of Incoming Handover (inHO) statistics in heterogeneous network. Simulation results showed that using the inHo data is more reliable than the traditional method using Manual Drive Tests (MDT) measurement data.

978-3-903176-15-7

The KNN and LOF algorithms for COD are compared in [13]. It is observed that KNN outperforms LOF in terms of speed and reliability since LOF can sometimes misclassify normal cells.

Diffusion maps (DM) is proposed by Chernogorov et al. [4] for dimensionality reduction whereby an unsupervised K-mean clustering algorithm is employed for further distinction between normal and abnormal data samples. The authors in [3] build upon [4] and use a dynamic clustering algorithm called Dynamic Affinity Propagation (DAP) algorithm to detect the anomaly cells. This DAP algorithm is better than the approach used in [2] and [4] because it does not require prior labeled training data as reference so as to save the storage of self-healing module.

Feng et al. [6] present a COD mechanism based on BP neural network to automatically detect outage cell and apply the differential evolution (DE) algorithm to improve the performance of standard Back Propagation (BP). The simulation results demonstrate that the proposed approach is more efficient and accurate compared with the standard BP algorithm.

The summary of the comparisons of self-healing detection and compensation are given in Table I and II, respectively. We give comparison of different solutions and their features. The method indicates the machine learning type. The detailed approach is described in sub_method. The performance metrics show the key performance information used in each approach. We also summarised both advantages and drawbacks of different approaches.

Based on the analysis in Tables I and II, we adopt the KNN classification as the detection algorithm because it is with high accuracy. At the same time, KNN is a lazy learning method that requires zero training time because the training instance is simply stored [14]. To be more exact, the KNN does not assume any underlying statistical model thus the training phase is fast and accurate thus

TABLE I: Summary of Self-healing detections

Solution	Method	Sub_method	Performance metrics	Advantages	Drawbacks
Outage Detection	Supervised learning	KNN[2]	RSRP, SINR	Simple, Accurate	Ambiguity in determining "k"
	Unsupervised learning	DAP[3]	RSRP, RSRQ, CQI	Automatically classify clusters	Quadratic computational cost
	Unsupervised learning	K-means [4]	RSRP, RSRQ, CQI, handover attempts	Simple	Ambiguity in determining "k"
	Unsupervised learning	LOF[5]	inHO	Simple	Misclassify normal cell
	Unsupervised learning	Neural Networks[6]	RSRP, RRC, DCR, HO, BCR	Accurate	Complicated

TABLE II: Summary of Self-healing compensations

Solution	Method	Sub_method	Performance metrics	Advantages	Drawbacks	Optimization objective
Outage Compensation	Heuristic	Immune Algorithm[7]	SINR	Fast compensation	Sensitive to initial parameters	Coverage and quality
	Heuristic	Genetic Algorithm[8]	Capacity	Immunity to initialization	Long time to converge with large data	Capacity unilization
	Unsupervised learning	Neural Networks[9]	Bandwidth signal, servers status signals	lower mean square error	Complicated	Spectral efficiency
	Reinforcement learning	Fuzzy-logic based RL[10]	SINR	Online learning	Complicated	Coverage

is suited for Wi-Fi network outage detection.

III. Proposed approach

In this section, we detail our proposed self-healing approach for Wi-Fi networks. The proposed approach leverages the ideas from Software Defined Network (SDN) by using an AP controller (APC) to collect the measurement data from all APs. The AP controller uses the OpenFlow 1.3 protocol [15] to communicate with individual APs.

Measured AP data is polled periodically and recorded in a table and stored the in APC. The neighbor APs information of each AP which can be obtained based the "Neighbor Report" mechanism specified in the IEEE 802.11k [16], [17] is saved in the table.

IV. Self-detection problem setup

In this section, a detailed description for the faulty AP detection algorithms are presented. We first present the problem formulation, and then introduce a supervised learning algorithm (KNN) and unsupervised learning (LOF) as solutions to the problem. Finally, we describe the evaluation criteria to compare the performance of KNN and LOF against the standard 802.11 performance.

A. Problem formulation

The aim of self-detection is to determine faulty APs and allow the APC to handover on-going connections to non faulty APs in the vicinity. This self-detection improves the QoS through minimising throughput degradation and packet loss during a handover.

Let $A_j(ap)$ denote a set of APs in the network on the jth TTI and $X_j(n)$ denote the measured data of each AP whereby AP is indexed by j such that $1 \leq j \leq n$ and $j \in \mathbb{Z}^+$. The set of $A_j(ap)$ and $X_j(n)$ are expressed as follows:

$$A_j(ap) = \{ap_{j,1}, ..ap_{j,k}, ...ap_{j,n}\}, ap_{j,k} \in N, \tag{1}$$

$$X_j(n) = \{x_{j,1}, ..x_{j,k}, ...x_{j,n}\}, \tag{2}$$

where $ap_{j,k}$ denotes the AP on the jth TTI with index k, $x_{j,k}$ denotes the measurement data of $ap_{j,k}$ and $x_{j,k}$ is a three-tuple defined as:

$$x_{j,k} = \{APID_s, energy_s, APID_{n1}\}, \tag{3}$$

where $APID_s$ denotes the serving AP, $energy_s$ denotes the remaining energy of the serving AP and $APID_{n1}$ denotes the first AP on neighbor list.

The real AP status include normal APs and faulty APs. Therefore, the detection function can be defined as follows:

$$y(x_{j,k}) = \begin{cases} \text{Normal} & \text{if } y(x_{j,k}) \leq \theta \\ \text{Faulty} & \text{if } y(x_{j,k}) \geq \theta \end{cases}, \tag{4}$$

where $x_{j,k}$ is the measurement data reported by each AP, threshold θ is used to learn the network behaviours when an AP fault occurs, the results of $y(x_{j,k})$ shows the AP status.

The detection problem can be used machine learning algorithm to predict the value $y(x_{j,k})$, thus the loss function can be defined as flows:

$$L(y(x_{j,k}), f(x_{j,k})) = \begin{cases} 0 & \text{if } y(x_{j,k}) \neq f(x_{j,k}) \\ 1 & \text{if } y(x_{j,k}) = f(x_{j,k}) \end{cases}, \tag{5}$$

where $f(x_{j,k})$ is the predicted value of AP status using machine learning algorithm and $y\,(x_{j,k})$ is the real value of AP status.

The machine learning algorithms for the faulty AP detection will be introduced in the following section.

B. Evaluation metrics

We use three metrics to evaluate machine learning algorithms performance. The metrics include Precision-Recall, $F1$ score and accuracy score.

Precision (P) is defined as the number of true positives (T_p) over the number of true positives plus the number of false positives (F_p). Recall (R) is defined as the number of true positives (T_p) over the number of true positives plus the number of false negatives (F_n). These quantities are also related to the (F_1) score, which is defined as the harmonic mean of precision and recall [18]: Precision (P) is defined as follows:

$$P = \frac{T_p}{T_p + F_p} \tag{6}$$

Recall (R) is defined as follows:

$$R = \frac{T_p}{T_p + F_n} \tag{7}$$

The F_1 score is defined as follows:

$$F1 = 2 \times \frac{R \times P}{R + P} \tag{8}$$

The accuracy score function computes the accuracy, either the fraction or the count of correct predictions. In multilabel classification, the function returns the subset accuracy. If the entire set of predicted labels for a sample strictly match with the true set of labels, then the subset accuracy is 1.0; otherwise it is 0.0 [18].

The fraction of correct predictions over n_{samples} is defined as:

$$\text{Accuracy}(y, \hat{y}) = \frac{1}{n_{\text{samples}}} \sum_{i=0}^{n_{\text{samples}}-1} 1_{\{\hat{y}_i = y_i\}}, \qquad (9)$$

where $\hat{y}_i$ is the predicted value of the i-th sample, y_i is the corresponding true value and 1 is an indicator function.

V. Evaluation

In this section, the performance of the proposed self-healing approach using KNN algorithm is evaluated compared with LOF and IEEE 802.11 standard.

A. Network setup

We implemented the proposed algorithms in ns-3 simulation.

TABLE III: Evaluation Parameters

Parameter	Value	Parameter	Value
Sim. time (t)	3000s	**STA Speed**	1m/s - 10m/s
Number of AP	12	**APs separation**	60m
Number of Nodes	1-35	**ActiveProbing**	True
Standard	802.11k	**PacketSize**	10000 byte
DataRate	50 Mbps	**AP power**	46.0206dBm

B. Analysis of Results

At t = 35s, AP3 was set to sleep status and transmit power was set to 0.1dBm to simulate hardware failures for 10 seconds. During the sleep mode, there was no energy consumption of AP3. We collected the remaining energy logs of all APs every 10 TTIs to make the temporal and spatial prediction.

1) Comparison of algorithms: In order to understand the KNN algorithm well, we also compare the different classification algorithms in Table IV. All three: KNN, support vector machine(SVM) and Random forest are supervised learning algorithms while K-mean and LOF are unsupervised learning algorithms. The results show that supervised learning can detects the abnormal APs with a higher accuracy – KNN has the best performance with accuracy at 99.28% and fast learning time of 0.01838s.

TABLE IV: Classifier comparison

Algorithms	F-measurement	Accuracy	Learning time
KNN	0.99	99.28%	0.01838s
SVM	0.98	98.38 %	0.10421s
Random forest	0.98	98.38%	0.03901s
K-mean	0.57	56.87 %	0.04387s
LOF	0.92	88.35%	0.02832s

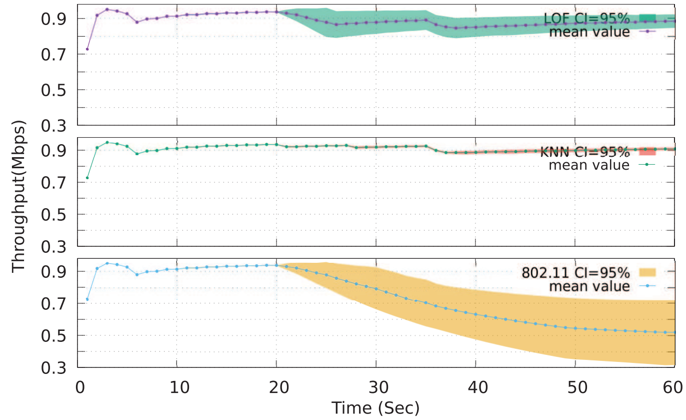

Fig. 1: Throughput for a typical 60s period.

2) Throughput: In Figure 1, we compared the throughput of KNN with LOF and 802.11 standard, all normalized to a maximum application level throughput of 50 Mbps. There is no self-healing mechanism in 802.11 standard. The STAs associated with faulty AP are disassociated when outage occurs in Wi-Fi networks decreasing the average throughput by 37.4% (labeled as 802.11 in Fig. 1).

With the self-healing approach, all the machine learning algorithms show the improvement of throughput. However, the average throughput of proposed KNN method has been improved by 63.6% compared with IEEE 802.11 standard while the average throughput of LOF only improved by 57.9%. This is because KNN detect the faulty AP faster than LOF and with high accuracy.

3) Packet loss rate: When the AP is faulty, it causes packet losses due to the momentary loss in connectivity. In the results shown in Figure 2, we record and compare the packet loss rate of KNN and LOF detection methods with 802.11 standard as the STA speed increases from 2m/s to 6m/s.

The traditional 802.11 standard shows very high average packet loss rate at 22.1%. After the self-detection and compensation for Wi-Fi networks, the average packet loss rate of KNN has been reduced to 5.5%. The packet loss rate of KNN is 16.6% lower than standard 802.11. The line of KNN is more stable than other algorithms such as SVM and random forest, which means the KNN method is not affected by the speed changing significantly. When the speed of STAs is slow, the SVM, random forest and LOF show good performance as KNN. However, when the speed of STAs increases, the accuracy of other three supervised learning algorithms decreases. Therefore, KNN is more suitable for the self-detection in different Wi-Fi network scenarios.

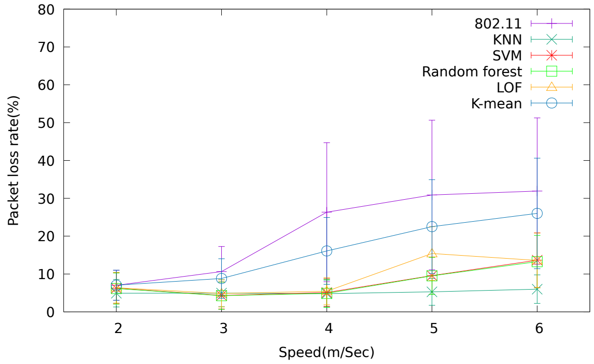

Fig. 2: Packet loss rate

4) Handover delay: Figure 3 shows the handover delay vs. the average number of handover recorded. When the number of handovers is low (1, 2 and handovers), all six mechanisms achieve almost identical handover delays, which is because there is no outage AP in the three handovers. As the STAs continues to move to AP3, the outage happens. As the number of handovers increases (4, 5 and 6 handovers), KNN, random forest and LOF achieve better handover performance. Although K-mean detects the faulty AP at the fourth handover, STAs lost connectivity after that because of lower detection accuracy rate. The fourth handover of 802.11 is after the outage restored. With five or more handovers, the 802.11 services is disconnected (hence no reading) while KNN achieves the lowest delay among the five machine learning algorithms.

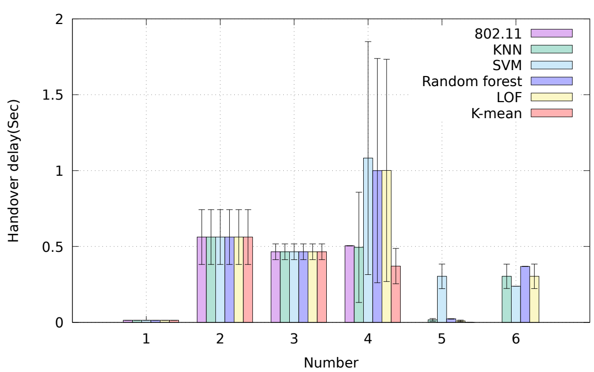

Fig. 3: Handover delay vs. average number of handovers in a 3000s period.

VI. Conclusion

This paper proposed a self-healing method for Wi-Fi networks to automatically detect the faulty APs and compensate the network performance.

References

[1] H. Gacanin and A. Ligata, "Wi-fi self-organizing networks: Challenges and use cases," *IEEE Communications Magazine*, vol. 55, no. 7, pp. 158–164, 2017.

[2] W. Xue, M. Peng, Y. Ma, and H. Zhang, "Classification-based approach for cell outage detection in self-healing heterogeneous networks," in *Wireless Communications and Networking Conference (WCNC), 2014 IEEE*. IEEE, 2014, pp. 2822–2826.

[3] Y. Ma, J. Zhu, L. Liu, F. Lv, and Y. Wang, "A dynamic affinity propagation clustering algorithm based on mdt in self-healing heterogeneous networks," in *Communications and Information Technologies (ISCIT), 2016 16th International Symposium on*. IEEE, 2016, pp. 318–323.

[4] F. Chernogorov, J. Turkka, T. Ristaniemi, and A. Averbuch, "Detection of sleeping cells in lte networks using diffusion maps," in *Vehicular Technology Conference (VTC Spring), 2011 IEEE 73rd*. IEEE, 2011, pp. 1–5.

[5] T. Zhang, L. Feng, P. Yu, S. Guo, W. Li, and X. Qiu, "A handover statistics based approach for cell outage detection in self-organized heterogeneous networks," in *Integrated Network and Service Management (IM), 2017 IFIP/IEEE Symposium on*. IEEE, 2017, pp. 628–631.

[6] W. Feng, Y. Teng, Y. Man, and M. Song, "Cell outage detection based on improved bp neural network in lte system," 2015.

[7] Z. Jiang, Y. Peng, Y. Su, W. Li, and X. Qiu, "A cell outage compensation scheme based on immune algorithm in lte networks," in *Network Operations and Management Symposium (APNOMS), 2013 15th Asia-Pacific*. IEEE, 2013, pp. 1–6.

[8] L. Xia, W. Li, H. Zhang, and Z. Wang, "A cell outage compensation mechanism in self-organizing ran," in *Wireless Communications, Networking and Mobile Computing (WiCOM), 2011 7th International Conference on*. IEEE, 2011, pp. 1–4.

[9] M. M. Al-Zawi, A. Hussain, D. Al-Jumeily, and A. Taleb-Bendiab, "Using adaptive neural networks in self-healing systems," in *Developments in eSystems Engineering (DESE), 2009 Second International Conference on*. IEEE, 2009, pp. 227–232.

[10] A. Asghar, H. Farooq, and A. Imran, "Self-healing in emerging cellular networks: Review, challenges and research directions," *IEEE Communications Surveys & Tutorials*, 2018.

[11] C. Kuan and K. Dimyati, "Analysis of collision probabilities for saturated IEEE 802.11 MAC protocol," *Electronics Letters*, vol. 42, no. 19, p. 1, 2006.

[12] ——, "Finite time-horizon markov model for IEEE 802.11 e," *Journal of Zhejiang University-SCIENCE A*, vol. 10, no. 10, pp. 1383–1388, 2009.

[13] A. Zoha, A. Saeed, A. Imran, M. A. Imran, and A. Abu-Dayya, "A son solution for sleeping cell detection using low-dimensional embedding of mdt measurements," in *Personal, Indoor, and Mobile Radio Communication (PIMRC), 2014 IEEE 25th Annual International Symposium on*. IEEE, 2014, pp. 1626–1630.

[14] S. B. Kotsiantis, I. Zaharakis, and P. Pintelas, "Supervised machine learning: A review of classification techniques," *Emerging artificial intelligence applications in computer engineering*, vol. 160, pp. 3–24, 2007.

[15] E. L. Fernandes and C. E. Rothenberg, "OpenFlow 1.3 software switch," *Salao de Ferramentas do XXXII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuıdos SBRC*, pp. 1021–1028, 2014.

[16] "Ieee standard: Radio resource measurement of wireless lans," *IEEE Std 802.11k-2008 (Amendment to IEEE Std 802.11-2007)*, pp. 1–244, June 2008.

[17] L. Hao, B. Ng, and Y. Qu, "Dynamic optimization of neighbor list to reduce changeover latency for wi-fi networks," in *Proceedings of the 2017 International Conference on Telecommunications and Communication Engineering*. ACM, 2017, pp. 20–24.

[18] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine Learning in Python ," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.