Performance Based Cells Classification in Cellular Network Using CDR Data

A. Rizwan, J. P. B. Nadas, M. A. Imran
School of Engineering
University of Glasgow, United Kingdom
[a.rizwan.1, j.battistella-nadas.1]@research.gla.ac.uk,
muhammad.imran@glasgow.ac.uk

M. Jaber
Smart IoT Research Group
Fujitsu Laboratories of Europe
London, UK
m.jaber@uk.fujitsu.com

***Abstract*—In the advent of ultra-dense networks with unprecedented complex and heterogeneous infrastructure, the role of automation in network optimization becomes vital for sustaining the target performance. In this work, we address the challenge of identifying and classifying sub-par performing nodes in near-real time through a machine-learning inspection of streaming performance indicators from multiple probe points. We present a novel K-means-based solution for classifying node performance over a sliding time segment and further categorizing the type of failure. The K-means solution first identifies the performance instances of interest. These are then inspected in a second clustering round for automated performance labeling. Next, the labeled data-set is employed to train a Support Vector Machine based classifier that is continuously classifying incoming performance instances from the network. The method is tested using a real network data set comprising call detail records. The results advocate the potential of our method for effectively and accurately identifying and classifying performance degradation in any node in the network.**

***Index Terms*—Clustering, Classification, Self-organizing Networks, K-Means, CDR data analysis, SVM.**

I. Introduction

The fifth generation of mobile networks (5G) is now a reality as Verizon turns on the world's first 5G network[1] and EE launches the first 5G trial in Canary Wharf, London[2]. The launch of 5G networks promises great gains in capacity, speed, resilience, and latency, but at the cost of unseen complexity. In order to unlock the 5G potential, a network requires ultra dense small cell deployments, multi-RAT coexistence (radio access technology), catering for an unprecedented spectrum of services, and advanced features (e.g., network slicing), just to name a few. Such complexity requires permanent monitoring, diagnosis, rectification, and actuation for three main reasons. First, it would require an inhibitive capital expenditure to be able to offer the promised 5G infinite capacity perception to all users by following the traditional over-engineering approach. To this end, network agility and flexibility are key to reshuffling the exactly dimensioned resources in "near" real-time in a user-centric method. Authors in [1] refer to this feature as "Resource Elasticity" and propose mechanisms for the exploitation of this elasticity by softwarising network functions pertinently and via cross-slice resource provisioning. Second, the increased number of network nodes coupled with the key role of each node in creating the infinite capacity perception renders the failure (or sub-par performance) of any node a significant impediment on the overall network performance. Thus, in order to offer the best sustainable quality of experience, an augmented smart root cause analysis with "near" real-time classification of network nodes is paramount. With that in mind, the authors in [2] propose a modified local outlier factor approach to identify cells in outage and apply rectification immediately in the context of heterogeneous cloud radio access networks. Third, the amount of data generated by the network to report on its performance and experience of its users is massive and very diverse in nature, content, and frequency. As such, it is essential to synthesize the knowledge from multiple sources in "near" real-time with permanent inspection and analysis to match the pace of changes in the network such as capacity demand, users expectations, interference conditions, mobility patterns, etc. Authors in [3] analyze the role of machine learning in the implementation of self-organized network management, with an end-to-end perspective of the network, taking into account the entirety of data generated by the network.

The methodology employed in current networks for performance management and optimization is by far insufficient to avail real-time optimization and requires a disruptive approach to scale it to 5G timing stipulations. Today's networks are firstly dissected to subsystems (e.g., radio access, transport, core, etc.), then to regions, and features (e.g, 3G, LTE, 4G, etc.) and sometimes to vendors (e.g., Nokia equipment, ZTE node B, Ericsson core, etc.) and more. Such a silo-like approach to examining the network performance is no longer valid in the world of 5G where the distinction between radio access and transport is blurred (e.g. C-RAN: centralized radio access networks with pooled base band units) and the inter-RAT operation is no longer optional (e.g., ATSSS: Access Traffic Steering, Switch and Splitting). Moreover, the abundance of data sources and data types has further increased with features such as minimization drive tests (MDT) [4] and advanced self organization mechanisms. There is an equal rise in specialized solutions to examine these new data sources, alas, these are often vendor specific and result in narrower

[1]https://www.techradar.com/uk/news/verizon-turns-on-the-worlds-first-5g-network

[2]https://www.mirror.co.uk/tech/ee-launches-first-5g-network-13368492

silos.

In this article, we propose a novel machine-learning approach for automating the detection of under-par network performance and a smart classification of the behaviour of network nodes which allows for a "near" real time detection of the nodes causing performance degradation. The described method is particularly designed to operate on various and multiple types of data streams that are continuously generated by the network. For instance, the performance indicators continuously generated by the radio access network may be jointly analyzed with minimization drive test data streams for the performance degradation detection. We posit that such a feature is crucial for overcoming the traditional silo-approach of network performance management.

The proposed solution offers a user-centric perception of the network's performance as opposed to the traditional network-centric perception. This aspect is critical in the prioritization of identified problems and in the distribution of resources/efforts for optimization and maintenance of network nodes. We have tested our solution on CDR data from an African GSM operator and it was successful in identifying under par performing nodes and categorize the "type" of degradation recorded within an hour of the incident. We thus summarize the contributions of our work in the following points:

- A new method for "near" real-time identification of under-par performing cells that may be applied to any network generated stream of data. Moreover, the identification is further categorized to highlight the "type" of degradation recorded in preparation for the automated smart root cause analysis that would follow.
- A new method for unsupervised learning based on K-means clustering that auto-tunes the pertinent number of clusters based on the streaming data.
- A supervised learning technique based on Support Vector Machines (SVM) that builds on the knowledge extracted from the previous step to classify similar new data streams.

The rest of the paper is organized as follows. Section II offers a survey on state of the art work that employs machine learning for solving network problems. Section III details the proposed method comprising two phases of clustering and classification. Sections IV and V describe the data set used for the validation of this method and the corresponding results and analysis. The paper is concluded in Section VI.

II. Machine Learning for mobile networks

Machine learning is a proven technique for solving complex problems and has been successfully applied in many fields such as computer vision, medical diagnosis, recommendation systems, speech recognition, and more. Driven by the success of machine learning in various verticals, both industry and the research communities are exploring its potentials in solving mobile network problems, as in [5].

There are three key features in machine learning that advocate for its application on mobile networks. First, machine learning learns from the data and the acquired knowledge improves when the data volume increases. In the advent of fast and massively parallel graphical processing units and the abundance of network-generated data, this key feature of machine learning is well exploited. Second, machine learning and reinforcement learning in particular, circumvents the requirement of highly complex closed form mathematical formulation since it is model-free and relies mostly on the reward system. Closed-form formulations have been a impediment in classical programming for mobile problems as they create a catch-22; many aspects of the system need to be omitted to secure a closed form, hence, compromising the model's fidelity, alternatively, no closed form can be reached which limits the solution to numerical analysis. The third features of machine learning is the knowledge transfer which, in the field of mobile networks, can exploit the temporal and spatial differences and relevance of different regions. As such, the knowledge acquired in one node can be transferred to another (or new) node to accelerate the learning curve. This is particularly important in the 5G era which is characterized by dynamic changes and diversity. In this case, the knowledge acquired in classifying macro-cell performance can readily be transferred to small cells or indoor cells. Similarly, knowledge acquired in diagnosing quality deterioration for VoIP services can be used to accelerate the diagnose of IoT service (e.g., e-health, smart meters, driver-less vehicles etc.).

Applying machine learning to solving network problems has resulted in a prolific research output in the last few years. Many efforts have been invested in identifying potential applications of machine learning such as [6]–[8] in the domain of wireless networks and in particular in the implementation of self-optimization mechanisms. Two recent works apply machine learning techniques into the analysis of Call Data Records (CDRs): [9] and [10]. Authors in [9] employ big data analysis for over ten million CDR records to extract the spatial-temporal predictability of network traffic. These CDR-driven predictions are then applied to a novel mechanism for joint optimization of energy consumption and inter-cell-interference in ultra-dense 5G networks. Motivated by the crippling cost of churn faced by telecom operators, authors in [10] apply deep learning to CDR and customer relationship management (CRM) records to predict which customer is likely to churn to allow for user-centric retention efforts. On the other hand, authors in [11] propose a novel method to automatically diagnose the radio condition in a mobile network cell based on the user's performance as captured by MDT records. To this end, the proposed method applies an unsupervised machine learning technique (Self optimizing maps) to cluster and classify the performance of each cell in the network. Moreover, in the domain of fixed networks, authors in [12] propose a machine learning method for performance monitoring employing SVM and double exponential smoothing (DES) for the prediction of equipment failure. Authors in [13] offer a solution that uses deep learning to predict customer churn. By exploiting the intrinsic property of deep neural networks, the proposed solution can be applied to any type of network and any subscription based events.

III. Methodology

In this section, we discuss the proposed solution, first detailing each step of the clustering stage then describing the SVM classifier. Steps taken to devise the solution are outlined in the flowchart presented in Fig. 1. The solution is devised with an aim to classify cells according to their performance in certain intervals of time over the day. By performance here we mean the volume of dropped calls (*i.e* duration, quantity) against that of normally terminated calls.

To this end, we start by analyzing the data to find relevant features that are descriptive of the desired performance aspect. Moreover, we apply the concept of sliding window to capture both, the instantaneous streaming values as well as the trend of variation of these values; the length of segments can be user defined, e.g., three hours. As there exists no gold standard for cluster validation in our case, therefore, in the clustering phase, the goodness of clusters is preliminarily determined by the internal measure of clusters compactness and separation. More importantly, we mostly rely on domain expertise via visual validation to determine the quality of the clusters and to ensure that domain specific characteristics are represented correctly. Tier-wise implementation scheme of K-means is designed with the approach mentioned in section III-C. A certain combination of features is selected for each tier after evaluating possible combinations from the available feature space.

Once the clustering is yielding satisfactory results data is labeled with the help of that clustering scheme. Then the labeled data is used to devise a classifier with supervised learning approach as it can be more simple and generic in terms of implementation on real data while being more deterministic in terms of performance evaluation at the same time. In the development of a classifier, the labeled data is split into training and testing data set. The first is employed to train the SVM classifier which is later evaluated on the testing data set. The entire process of clustering and classification is reiterated until a satisfactory level of accuracy is achieved resulting in a final clustering and classification scheme.

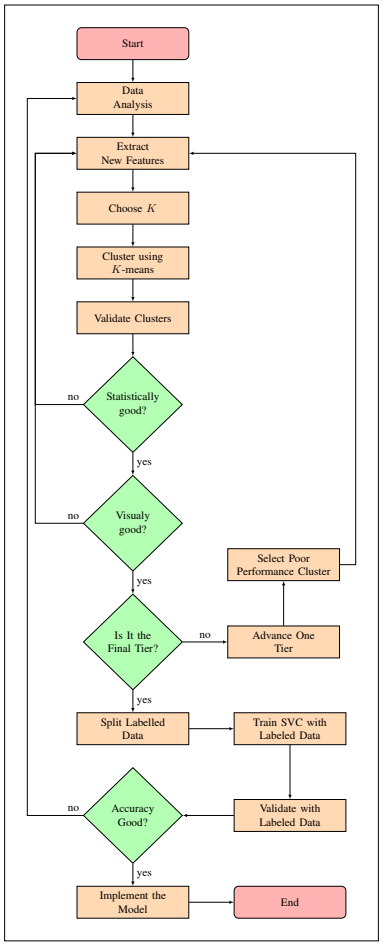


Fig. 1. Classification Process Flow Chart

A. Clustering Model

The goal here is to identify categories of issues present in network and group together segments with similar performance behavior. As, here, different possible categories of performance behavior are not known, so, the use of a clustering algorithm is an obvious choice. But the fact is that there is no best clustering algorithm [14]. Some of the major methods of clustering are based on density estimation, probabilistic estimation, partition, and graph-theoretic.

There are numerous clustering algorithms which basically differ in their objective function computation. Each algorithm has its own pros, cons and implementation methods. Density-based clustering algorithms [15] e.g., DBSCAN that looks for regions of high density is not very efficient particularly for high dimensional sparse data cases because it compares the distance of all pairs of points.

Whereas, though the spectral clustering [16], an example of graph theory, does not require complicated parameters like BDSCAN, but it is not a good algorithm for big data and a higher number of clusters. A Gaussian mixture model [17] from the family of probabilistic methods consider data as a mixture of Gaussian distribution with unknown factors. This model has limitations like it cannot be scaled and it requires too many parameters for implementation.

K-means clustering is the most popular and simplest partition model [18]. It does not require any parameters except the number of clusters. There also exist the methods to compute an optimal number of clusters. It works well even for high-dimensional sparse data [14]. Multiple variants and implementation schemes of K-means exist due to the rich history of research [14]. In addition, it is scale-able for huge data and very efficient for real-time implementation. Considering all these factors K-means clustering is selected here.

B. Clustering Validation Metrics

Another critical task in clustering is the validation of results, particularly in absence of a ground truth, as it is the case here. We have no predefined labels for any classes in our data. There exist multiple metrics for clustering validation, for data, where exists ground truth or golden standard, also known as external

metrics [19]. But the metrics for evaluating the goodness of clustering without ground truth are rare. Such metrics are called internal metrics and they evaluate clusters generally on the basis of two attributes of clustering: compactness within clusters and separation between clusters [20]. Some of the internal metrics take compactness into consideration or separations only, other evaluate clustering taking both parameters into consideration at the same time. We have taken four commonly used and well-acknowledged internal metrics [20]: Root-mean-square standard deviation (RMSSTD) for compactness, R-squared (RS) for separation, Calinski-Harabasz index (CH) and Silhouette index (S) for compactness and separation.

Internal metrics basically help to determine the optimal number of clusters and their numeric values just indicate how compact or separate the clusters are on average. For example, Silhouette index varies between [-1,1] where negative values indicate that clusters are mixed with data points assigned to one cluster from other clusters. When it is zero or close to zero, it reflects clusters are not far from each other and similarity among data points within clusters is low. Moving away from zero on the positive side of the index indicates that the clusters are well separated and well compact. Besides that, visual validation with the help of domain knowledge is a requirement in the absence of a gold standard. These metrics can be a good statistical indicator of compactness or separation of the cluster but they are no guarantee that the obtained clusters are suitable for the desired application [20].

C. Clustering Implementation Scheme

K-means clustering results in groups of data such that a distance metric between the empirical mean of a sub-cluster and the points in that cluster is minimized and it happens for all the clusters [14]. In this work, we have used the Euclidean distance as the distance metric. The input parameter required by K-means is the number of clusters beside decision function also known as kernel.

In this research, an heuristic approach is applied for the selection of features. For this different combinations of features are used in K-means and the features with optimal results are shortlisted. Clusters here not only need to be statistically sound but require to group together segments with similar performance in terms of telecommunications characteristics, that is variations of cell load alongside variations in the number of interrupted calls. Therefore cluster validation is done by applying domain knowledge on the visual presentation of segments in clusters alongside the use of internal validation metrics. Moreover, the selection of clustering scheme is another key factor that affects the results. The data has multiple features and co-relations among those features and variation within the values of individual features determine the network behavior. We have applied a two-tier clustering scheme, wherein the first tier separates the segments with poor performance from those with good performance while the second tier further segregates the poor performing segments into the final clusters reflecting different type of network behaviors. Another advantage of using this two-tier approach is that visual validation is easier as there are fewer segments to analyze in the second tier.

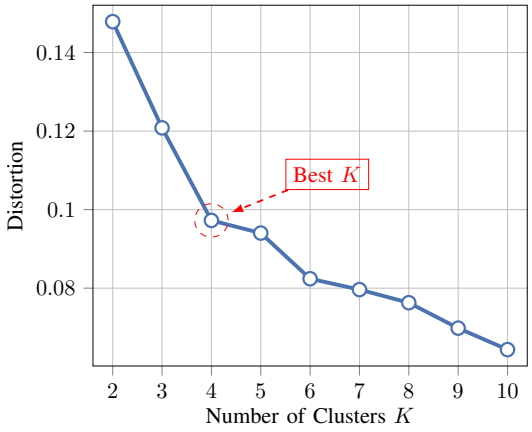


Fig. 2. Elbow method considering the first tier for the selection of K.

In both tiers, we majorly rely on the elbow method beside the metrics of compactness and separation for selecting the optimal K. Bend on the elbow plot as shown in Fig. 2 is the optimal number as after that increase in number of clusters does not changes the clustering at that rate as it change it before that, similar behavior is observed when RMSS and RS values in Table I are plotted against the number of clusters. Calinski-Harabasz index (CH) and Silhouette index (S) are used to evaluate the overall quality of the clusters.

The inputs of each tier are the hourly aggregated CDRs, split into l long vectors for each segment i and cell j. $\vec{D}_{i,j}$ and $\vec{D}'_{i,j}$ are the duration of normally terminated and dropped calls, while $\vec{C}_{i,j}$ and $\vec{C}'_{i,j}$ express the quantity of normally terminated and dropped calls, respectively. Moreover, the vectors $\vec{D}_j$, $\vec{D}'_j$, $\vec{C}_j$ and $\vec{C}'_j$, with length 24, contain data relative to cell j for the entire day.

1) First Tier: Tier I is used to separate segments with poor performance from the segments with good performance in terms of number and duration of interrupted calls. It also segregates segments with good performance according to the cell load. For that we have used two features F^1 and F^2 computed as follows:

$$F^1_{i,j} = \text{mean}\left(\frac{\vec{D}_{i,j}}{\max\left(\vec{D}_j\right)}\right) \tag{1}$$

and

$$F^2_{i,j} = 10 \cdot \log\left(\max\left(\vec{C}'_{i,j} \div \left(\vec{C}_{i,j} + \vec{C}'_{i,j}\right)\right)\right), \tag{2}$$

where $\div$ indicates an element wise division and $\log$ denotes the natural logarithm.

To emphasize differences in the number of dropped calls, we have considered a logarithmic scale for F^2. The idea is to make the variation of the bad data more pronounced, such that clustering can find the bad quality segments easily based on Euclidean distance. The output of the first tier is depicted

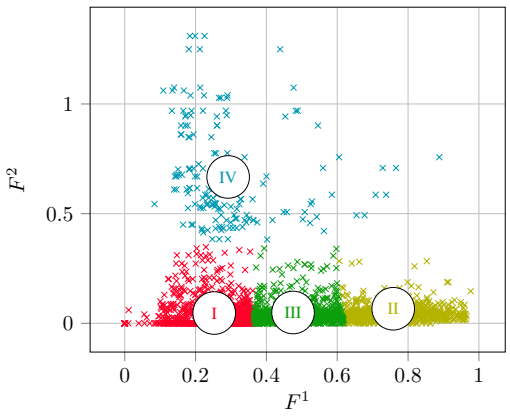


Fig. 3. Clusters at Tier 1

in Fig. 3. Note that the bad quality segments are grouped into Cluster IV while the other three clusters contain good quality segments segregated according to their volume of traffic.

2) Second Tier: In the second tier, we consider only the segments from Cluster IV, the one with poor performing segments from the first tier. It is further sub-clustered into three groups of segments with K-means clustering. There are two sets of features used in the second tier. The first set $F^3_{i,j}$ is obtained via

$$F^3_{i,j} = \nabla \left(\frac{\vec{C}_{i,j}}{\max\left(\vec{C}_j\right)} \right), \tag{3}$$

where ∇ indicates the gradient operation and thus $F^3_{i,j}$ is a vector of $l-1$ features.

The second set $F^4_{i,j}$ is obtained using

$$F^4_{i,j} = \nabla \left(\vec{D}'_{i,j} \div \left(\vec{D}_{i,j} + \vec{D}'_{i,j} \right) \right), \tag{4}$$

and thus also results in a vector of $l-1$ features. Therefore the second tier contains $2l-2$ features in total. $F^3_{i,j}$ captures the variation in the number of bad calls as a gradient whereas $F^4_{i,j}$ encompass variations in bad call duration.

D. Classification

The labeled data from K-Means clustering is used further to train and test a classifier. Here an SVM based classifier (SVC) is used to classify the labeled data. SVM is very effective for high dimensional sparse data even for the cases where the number of dimensions are greater than the number of samples. It is memory efficient for training as it uses a subset of points for the training of the decision function. SVM based models already have proved to perform exceptionally on cellular network data like the call traffic prediction at high granularity [9]. Another huge advantage of SVC is the availability of a diverse range of Kernels to compute the decision function [21], such as linear, polynomial of higher degrees, Gaussian, sigmoid etc. In this research, we have evaluated the three most popular kernels, linear, polynomial (cubic) and radial basis function (RBF) for developing the classifier. SVC decision function constructs hyper-plane(s) in high dimensional space which separates the data points into different classes. Theoretically, good separation is when the hyperplane has maximum distance from the nearest training data points of any class. But there is a trade-off between the separation level also known as functional margin and model generalization. The higher the margin, the lower the generalization error of the classifier leading to over-fitting. Model generalization and over-fitting can be regulated with the help of kernel parameters like C also known as penalty parameter for the error. A larger C yields a more accurate classification at the cost of a lower generalization. A smaller C means a more smooth decision plane. We have used a set of exponentially increasing values for C for all three kernels. For training the classifier, the data is randomly split into training and testing data sets, such that 75% is allocated for training and validation while the remainder 25% of the data is reserved for testing. A classifier is trained and validated using 75% of the training data with different combinations of kernels and C. Three-fold cross-validation is applied on the training data with all parametric schemes. The best estimated model is then applied on the test data in the end for the final evaluation of classifier.

IV. Data-set Description

The data used in this research is extracted from CDRs gathered from a real GSM network in Africa over a period of three weeks for 759 geographical cells. It contains information regarding the duration and quantity of dropped calls and calls terminated as normal. Normal termination means the calls were properly cut off by either user, conversely, a dropped call is a call that has ended due to some network error. We have used hourly aggregated data of 759 cells for one day to train and evaluate our clustering and classification scheme.

From our preliminary analysis, we found that total call traffic in terms of duration and quantity is very low from midnight until morning as shown in the Fig. 4, which contains the average[3] total traffic across all the cells for one day. Thus, we have taken the data from 6 a.m. to midnight into consideration for this research. Selecting the data this way also helps to overcome the model biases towards greater number of segments with low traffic. Moreover, network operators are more concerned in dealing with issues which affect larger volumes of calls, which occur during the day.

V. Results

A. Clustering

1) Tier I: From the results of elbow method as shown in Fig.2 and those of RMSS and RS shown in table I it is found that $K = 4$ is the optimal number of cluster at Tier I. Using $K = 4$ for the feature sets F^1, F^2 produces optimal clusters presented in Figs. 3. The use of a logarithmic scale for the dropped calls in F^2 helps to capture smaller variations in

[3]The traffic is first scaled between 0 and 100 for the whole day and each cell, then averaged. This provides a realistic information of the cell load at each hour of the day.

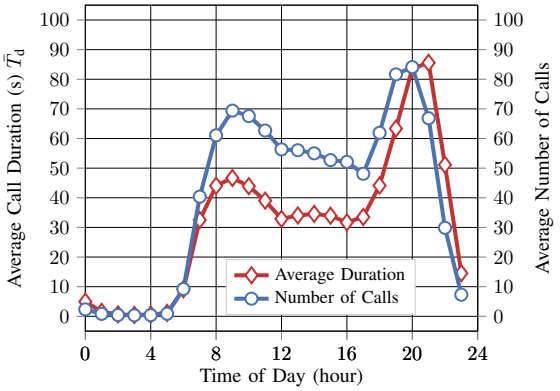


Fig. 4. Average normalized traffic

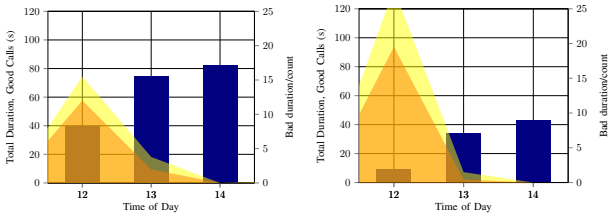
dropped calls volume, and therefore segregates the segments more meaningfully. As a result of clustering at Tier I four clusters obtained comprise segments with good quality and low traffic in Cluster I, medium traffic in Cluster II, high traffic in Cluster III. Cluster IV contains segments with bad quality traffic.

2) Tier II: At this tier Cluster IV with bad quality traffic segments obtained as the result of Tier I clustering is further segregated into sub-clusters on the basis of variation in performance. Based on the statistical and visual validation features set F^3, $F^4$are found to produce the best results for $K = 3$ on Tier II. Statistical results for tier two with the optimal feature combination are presented in Table I. These numbers indicate how compact or separated the clusters are at Tier II. Where RS and RMSSD scores respectively give an idea about how separate or compact the clusters are there they also help to choose an optimal number of clusters which is III for Tier II.

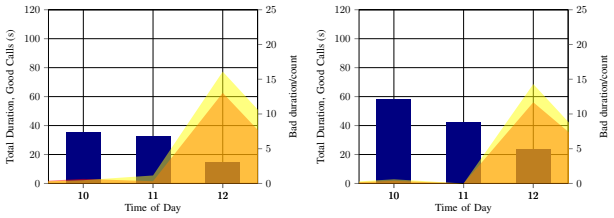
The results are found to be very promising as shown in Figs. 5a-5c presenting samples from each cluster at Tier II. Red color represents the duration percentage of dropped calls while the yellow color represents the percentage of dropped calls (quantity), both presented on the right y-axis, whereas normalized good calls duration is presented by blue bars on the left y-axis. Figure 5a represents Cluster IV, it can be seen that segments with decreasing dropped calls and increasing good call duration are grouped in that cluster. Whereas segments

TABLE I
CLUSTERING VALIDATION RESULTS

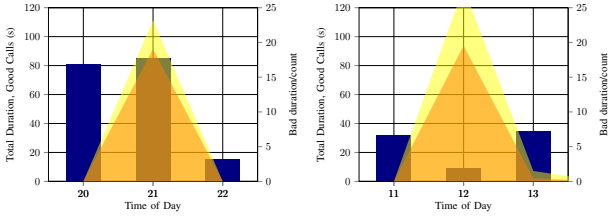
K	RMSS		RS		Silhoutte		CH	
	T-1	T-2	T-1	T-2	T-1	T-2	T-1	T-2
2	0.14	0.18	0.44	0.33	0.48	0.36	9546	312
3	0.11	0.15	0.69	0.54	0.54	0.39	13479	368
4	0.09	0.14	0.78	0.58	0.45	0.37	13963	296
5	0.08	0.14	0.81	0.62	0.46	0.38	13206	262
6	0.07	0.13	0.85	0.66	0.41	0.36	13383	246
7	0.07	0.13	0.87	0.69	0.41	0.31	13126	232
8	0.06	0.12	0.88	0.71	0.42	0.31	13278	223
9	0.06	0.12	0.90	0.73	0.41	0.32	13910	217
10	0.06	0.11	0.91	0.76	0.39	0.29	14052	220



(a) Cluster IV: Recovering from performance degradation.



(b) Cluster V: Beginning of performance degradation.



(c) Cluster VI: Unstable performance.

Fig. 5. Sample segments in their respective clusters.

with an opposite behavior, to that observed for Cluster IV, are grouped in Cluster V as shown in Fig.5b. Fig. 5c shows segments in Cluster VI which are different from IV and V in the traffic patterns, here we can see a spike in bad calls on the middle hour when the good traffic is low compared to other hours.

TABLE II
CLASSIFICATION CROSS VALIDATION RESULTS

Rank	C	Kernel	Mean Validation Score	Mean Train Score
1	10^3	Linear	0.989	0.99
2	10^4	Linear	0.986	1.00
3	10^2	Linear	0.986	0.99
4	10^2	RBF	0.985	0.99
5	10^3	RBF	0.984	0.99
6	10^4	RBF	0.984	1.00
7	10^1	Linear	0.983	0.99
8	10^4	Cubic	0.975	0.98
9	10^1	RBF	0.974	0.98
10	10^0	Linear	0.970	0.97
11	10^3	Cubic	0.961	0.97
12	10^2	Cubic	0.931	0.93
13	10^0	RBF	0.923	0.92
14	10^{-1}	Linear	0.914	0.91
15	10^1	Cubic	0.877	0.88
16	10^{-1}	RBF	0.862	0.86
17	10^0	Cubic	0.797	0.80
18	10^{-1}	Cubic	0.581	0.58

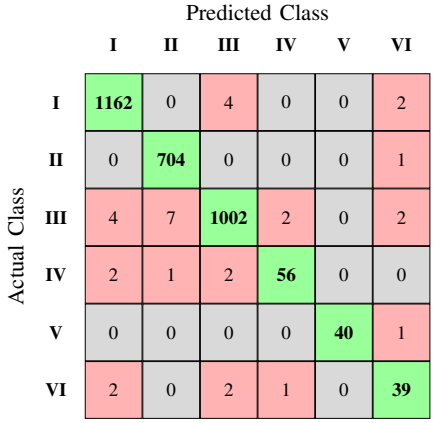


Fig. 6. Confusion Matrix

B. Classification

Labeled data of six clusters produced from the both tiers of clustering is used to train, validate and test the SVC. Results of three-fold cross-validation are shown in Table II which are very promising not only in term of accuracy but also for the persistence of the model. It can be seen from the mean training and validation score for different sets of parameters shown in Table II that the model performance is consistent, reducing the chances of over-fitting. Best mean accuracy score for cross-validation is 99.39% for $C = 1000$ and Linear Kernel i.e. more than 99% of the segments are assigned accurately to their classes. The classifier trained for these parameters yields 98.91 % accuracy for unseen test data. It is also evident from the confusion matrix in Fig. 6 that the classifier is not only able to correctly identify the segments from the large clusters but also from the smaller clusters, in Cluster IV 92%, in Cluster V 98 %, and in Cluster VI 88% of the segments are identified correctly. It also reflects that the model is not biased towards clusters with more segments.

VI. Conclusion

As we approach the roll-out of 5G and embark on the design of 6G, the challenge of identifying and classifying node with performance degradation is increasing rapidly while the key role of each node in delivering the quality of experience to the users is becoming more evident. In this work, we present the first machine-learning-based automated solution for filtering out performance degradation and classifying the type of deterioration in near-real time from streaming network-generated metrics. We validate the method using CDR data from a real network and demonstrate its benefits as it yields 98.91 % accuracy in classification and spares precious time of domain experts to fix the problem as opposed to identify and diagnose it. This work can be extended to multiple performance indicator streams and hence can unlock the limitations created by the traditional silo-approaches. Besides that, it can be extended towards a more proactive approach that predicts the possible network behavior on the basis of previous patterns.

References

[1] D. M. Gutierrez-Estevez, M. Gramaglia, A. de Domenico, N. di Pietro, S. Khatibi, K. Shah, D. Tsolkas, P. Arnold, and P. Serrano, "The path towards resource elasticity for 5G network architecture," in *2018 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, April 2018, pp. 214–219.

[2] P. Yu, F. Zhou, T. Zhang, W. L. L. Feng, and X. Qiu, "Self-organized cell outage detection architecture and approach for 5G H-CRAN," *Wireless Communications and Mobile Computing*, vol. 2018, no. Article ID 6201386, p. 11, 2018.

[3] J. Moysen, L. Giupponi, and J. Mangues-Bafalluy, "A mobile network planning tool based on data analytics," *Mobile Information Systems*, vol. 2017, no. Article ID 6740585, p. 16, 2017.

[4] 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.

[5] F. Pervez, M. Jaber, J. Qadir, S. Younis, and M. A. Imran, "Memory-based user-centric backhaul-aware user cell association scheme," *IEEE Access*, vol. 6, pp. 39 595–39 605, 2018.

[6] M. Chen, U. Challita, W. Saad, C. Yin, and M. Debbah, "Machine learning for wireless networks with artificial intelligence: a tutorial on neural networks," *arXiv preprint arXiv:1710.02913v1*, 2017.

[7] P. V. Klaine, M. A. Imran, O. Onireti, and R. D. Souza, "A survey of machine learning techniques applied to self-organizing cellular networks," *IEEE Communications Surveys Tutorials*, vol. 19, no. 4, pp. 2392–2431, Fourthquarter 2017.

[8] Y. Sun, M. Peng, Y. Zhou, Y. Huang, and S. Mao, "Application of machine learning in wireless networks: key techniques and open issues," *arXiv preprint arXiv:1809.08707v1*, 2018.

[9] A. Zoha, A. Saeed, H. Farooq, A. Rizwan, A. Imran, and M. A. Imran, "Leveraging intelligence from network CDR data for interference aware energy consumption minimization," *IEEE Transactions on Mobile Computing*, vol. 17, no. 7, pp. 1569 – 1582, Nov 2018.

[10] F. Castanedo, G. Valverde, J. Zaratiegui, and A. Vazquez, "Using deep learning to predict customer churn in a mobile telecommunication network," *http://wiseathena.com*, 2016.

[11] A. Gómez-Andrades, R. Barco, P. Muñoz, and I. Serrano, "Data analytics for diagnosing the RF condition in self-organizing networks," *IEEE Transactions on Mobile Computing*, vol. 16, no. 6, pp. 1587–1600, June 2017.

[12] P. Yu, F. Zhou, T. Zhang, W. L. L. Feng, and X. Qiu, "Failure prediction using machine learning and time series in optical network," *OPTICS EXPRESS 18553*, vol. 25, no. 16, August 2017.

[13] P. Spanoudes and T. Nguyen, "Deep learning in customer churn prediction: unsupervised feature learning on abstract company independent feature vectors," *arXiv preprint arXiv:1703.03869v1*, 2017.

[14] A. K. Jain, "Data clustering: 50 years beyond K-means," in *Machine learning and knowledge discovery in databases*. Springer, 2008, pp. 3–4.

[15] H.-P. Kriegel, P. Kröger, J. Sander, and A. Zimek, "Density-based clustering," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 1, no. 3, pp. 231–240, 2011.

[16] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: analysis and an algorithm," in *Advances in neural information processing systems*, 2002, pp. 849–856.

[17] C. Biernacki, G. Celeux, and G. Govaert, "Assessing a mixture model for clustering with the integrated classification likelihood," Ph.D. dissertation, INRIA, 1998.

[18] C. Ding and X. He, "K-means clustering via principal component analysis," in *Proceedings of the twenty-first international conference on Machine learning*. ACM, 2004, p. 29.

[19] R. J. Campello, "Generalized external indexes for comparing data partitions with overlapping categories," *Pattern Recognition Letters*, vol. 31, no. 9, pp. 966–975, 2010.

[20] M. Hassani and T. Seidl, "Using internal evaluation measures to validate the quality of diverse stream clustering algorithms," *Vietnam Journal of Computer Science*, vol. 4, no. 3, pp. 171–183, 2017.

[21] B. Scholkopf, K.-K. Sung, C. J. Burges, F. Girosi, P. Niyogi, T. Poggio, and V. Vapnik, "Comparing support vector machines with Gaussian kernels to radial basis function classifiers," *IEEE transactions on Signal Processing*, vol. 45, no. 11, pp. 2758–2765, 1997.