# Enabling Proactive Self-Healing by Data Mining Network Failure Logs

Umair Sajid Hashmi, Arsalan Darbandi, Ali Imran
The University of Oklahoma
Department of Electrical and Computer Science, Tulsa, OK
Email: {umair.hashmi, arsalan.darbandi, ali.imran}@ou.edu

***Abstract*—Self-healing is a key desirable feature in emerging communication networks. While legacy self-healing mechanisms that are reactive in nature can minimize recovery time substantially, the recently conceived extremely low latency and high Quality of Experience (QoE) requirements call for self-healing mechanisms that are pro-active instead of reactive thereby enabling minimal recovery times. A corner stone in enabling proactive self-healing is predictive analytics of historical network failure logs (NFL). In current networks NFL data remains mostly dark, i.e., though they are stored but they are not exploited to their full potential. In this paper, we present a case study that investigates spatio-temporal trends in a large NFL database of a nationwide broadband operator. To discover hidden patterns in the data we leverage five different unsupervised pattern recognition and clustering along with density based outlier detection techniques namely: K-means clustering, Fuzzy C-means clustering, Local Outlier Factor, Local Outlier Probabilities and Kohonen's Self Organizing Maps. Results indicate that self-organizing maps with local outlier probabilities outperform K-means and Fuzzy C-means clustering in terms of sum of squared errors (SSE) and Davis Boulden index (DBI) values. Through an extensive data analysis leveraging a rich combination of the aforementioned techniques, we extract trends that can enable the operator to proactively tackle similar faults in future and improve QoE and recovery times and minimize operational costs, thereby paving the way towards proactive self-healing.**

***Index Terms*—K-means clustering, Fuzzy C-means clustering, Self Organizing Maps, Local Outlier Factor, Local Outlier Probabilities, Network Failure Log database**

## I. Introduction

As the global mobile data increased to 3.7 exabytes per month at the end of 2015, 51% of that data was offloaded onto the fixed infrastructure [1]. With an increase in femto cell deployment density and advent of Internet of Things (IoT) as one of 5G use cases, maintenance and reliability of existing broadband infrastructure is key to sustaining the data requirements. On the other hand, gradually decreasing average revenue per user (ARPU) and the cost of reliable backhaul for small cells is a growing a pain point for mobile operators [2]. These trends translate into need for providing reliable broadband, while keeping operational costs low. To meet this pressing need more intelligent mechanisms to optimize, maintain and troubleshoot the broadband infrastructure have to be developed. One possible approach to achieve these objectives is to exploit the massive amount of data that can be harnessed from the network. Systematic analysis of such big data can be leveraged to minimize operational cost, maximize resources efficiency, and enhance customers' quality of experience (QoE).

Inspired by the network telemetric data exploitation framework presented in [3], in this paper, we present findings of our comprehensive analysis of a real network failure log (NFL) data set obtained from a nationwide broadband service provider serving about 1.3 million customers. The data is extracted from company's Siebel customer relationship management (CRM) system that records and tracks status of customer complaints along with network generated alarms that affect a particular region during certain time. The selected data spans duration of 12 months and contains about 1 million NFL data points from 5 service regions of the company. The extracted data has 9 attributes out of which 5 are selected for our analysis. These analyzed attributes include: 1) fault occurrence date, 2) time of the day, 3) geographical region, 4) fault cause and 5) resolution time. The objective of the study is to convert this raw NFL data, into a knowledge base that can readily be used by the operator to take more optimal decisions for minimizing operational cost, minimizing recovery time and maximizing QoE.

**Problem Statement:** To this end, this paper serves to investigate the following hypotheses:

$H_o$: We can identify clusters with distinct spatio-temporal features within the NFL data set by applying data mining techniques.

$H_1$: Out of the proposed techniques, there exists one or a combination of multiple machine learning techniques that provide optimal clustering and anomaly detection results.

To perform this analysis, instead of taking the classic statistical approach, where a sample of the data is analyzed to draw conclusions that are then extrapolated for the whole data, we take big data based approach in which the whole of data is analyzed without any sampling. While the former approach can help reduce the number erroneous entries through careful selection of samples from the whole data, the advantage of the big data based approach is that it can bring forth subtle patterns and insights which can be missed by sampling based approach. To explore hidden patterns in the NFL data, and to see which machine learning tools yields best insights, we apply a range of unsupervised learning and density based local outlier analysis techniques namely: 1) K-means clustering, 2) Fuzzy C-means (FCM) clustering, 3) Kohonen's Self Organizing Maps, 4) Local Outlier Factor (LOF) and 5) Local Outlier

978-1-5090-4588-4/17/$31.00 ©2017 IEEE

Probabilities(LoOP). The results of these different algorithms are compared in terms of sum of squared errors (SSE) and Davis Boulden index (DBI) values. Since the NFL data is unlabeled, unsupervised clustering techniques are preferred over supervised clustering as they provide unbiased groups of similar NFL traits. The applied techniques are established in the literature for efficient unsupervised clustering and anomaly detection in clustered data which makes them suitable for our study.

The novel insights revealed by the presented analysis can not only be used for minimizing the maintenance costs, but also to improve the QoE by minimizing the recovery time. Minimization of recovery time is possible through the presented NFL analysis, because by building on spatio temporal trends of certain or all network failures revealed by this analysis, a proactive instead of reactive maintenance schedule can be designed. The rest of the paper is organized as follows. Section II presents a review of relevant literature work, in section III we introduce the machine learning techniques, section IV elaborates on the data attributes and pre-processing techniques to normalize the NFL data, and in section V the learning results and key insights are discussed followed by the conclusions in Section VI.

## II. Related Work

Big Data empowered Self Organizing Networks (BSON) can be leveraged to utilize the huge amount of network information and create end-to-end visibility of the network resulting in improved quality of service (QoS) [3]. For instance, one of the exciting trends in application of machine learning algorithms on network generated data is the analysis of call data records (CDRs). The authors in [4] explained the application of K-means clustering for anonymized CDRs to find usage groups and optimal clusters for their datasets. In [5], K-means algorithm was applied on the CDRs of the employees of an IT company to form 4, 6 and 8 clusters to identify trends such as voice calls, SMS, call durations and data traffic. On a general basis, it is seen that employee level could be identified based on the cluster assigned, for example the top management of the company was assigned a single cluster due to similar pattern of CDRs. [6] discussed different machine learning (ML) methodologies for customer churn prediction in telecom industry. The initial results are compared with performance enhancement using boost methodology. The authors used churn data from UCI Repository and applied ML techniques and classification methodologies such as ANN, support vector machines (SVMs), DTs, Naïve Bayes classifiers and logistic regression. Another useful perspective on application of different algorithms for churn prediction in telecommunication industry using K-means clustering was presented in [7].

Literature also provides examples of clustering for customer satisfaction using different attributes of telecom users such as [8] where the authors gave a detailed account of hierarchical cluster analysis through its different techniques; top down (divisive approach) versus bottom up (agglomerative approach) to create the clusters with similar traits. A comparison of K-means algorithm with fuzzy C-means is given in [9] where the authors performed clustering on 4 attributes of broadband service on about 285,000 data points. Results indicated that although K-means algorithm is computationally efficient, C-means is more prone to noise in the data. Self organizing maps (SOM) provide higher classification accuracy as compared to K-means clustering for a variety of synthetic and real world datasets [10]. Classification accuracy for SOM is found to be superior for lower number of clusters; however as the number of clusters increases, K-means clustering shows similar performance [11].

Compared to existing studies, the novelty of this paper is two folded: first, to the best of our literature survey, real NFL data of this size and nature has not been analyzed before and second, this is a first study to compare the performance of K-means and fuzzy C-means with SOM by leveraging LOF and LoOP analysis on same real data set. Through our analysis in this study, we propose using proactive self-healing schemes to minimize number of service outage events and mean outage duration. For a review on possible self-healing frameworks based on network generated big data, please refer to our recent work in [12][13][14][15].

## III. Employed Algorithms

### A. K-means clustering

K-means is a prototype-based partitional clustering technique that clusters the given data in K clusters where K is the user-specified number of clusters [16]. It is the most commonly used clustering technique that creates one-level partitioning of a continuous n-dimensional data with centroid-based prototyping. The centroid assignment and updation cycle is repeated until the centroids remain very similar or there is a negligible percentage of data points changing clusters. The optimal cluster centroid which minimizes the SSE is the mean of all the data points assigned to the cluster and given by

$$c_k = \frac{1}{m_k} \sum_{x \varepsilon c_k} x_k. \tag{1}$$

In our analysis, we employ Elbow method that determines K as the point when decrease in SSE becomes linear as we increase K incrementally. To avoid sub-optimal clustering, we choose random centroids multiple times and select the set of centroids that gives us minimum initial SSE. Our post-processing technique alternately splits and merges K-means clusters so that the SSE is reduced but the total number of clusters remains fixed. The K-means algorithm applied is summarized in Algorithm 1.

### B. Fuzzy C-means (FCM) clustering

In the traditional K-means algorithm, a data point belongs to a set with certainty of either 0 or 1. In FCM, each data point $x_i$ is assigned a degree of membership with each cluster $C_j$ through a membership weight $w_{ij}$ that varies between 0 and 1 [16]. The weights for data points sum to 1 and each cluster contains at least one point with a non-zero weight,

**Algorithm 1** K-means clustering algorithm

1: Randomly select K points multiple times as cluster centroids and select the ones with minimum SSE.
2: **Repeat**
3: Cluster dataset by calculating minimum distance of each data point with all K centroids.
4: Recompute the centroid of each cluster.
5: **Until** centroids do not vary above a fixed percentage.

i.e. $\sum_{j=1}^{k} w_{ij} = 1$ and $0 < \sum_{i=1}^{n} w_{ij} < m$. Like K-means, FCM aims to minimize the SSE using centroid updation and assigning each data point to the closest centroid calculated using (3). The SSE calculation is measured on Euclidean ($L^2$) distance multiplied by $w_{ij}$ for each cluster,

$$c_j = \frac{\sum_{i=1}^{n} w_{ij}^{p} x_i}{\sum_{i=1}^{n} w_{ij}^{p}}. \quad (2)$$

$p$ represents the rate of weight in (3) and can have any value between 0 and 1. The FCM algorithm is summarized below as Algorithm 2.

**Algorithm 2** Fuzzy C-Means clustering algorithm

1: Assign membership weights to each data point based on minimum overall SSE.
2: **Repeat**
3: Compute the centroid of each cluster.
4: Recompute the membership weights of data points.
5: **Until** centroids do not vary above a fixed percentage.

### C. Kohonen's Self Organizing Maps (SOM)

Kohonen Self-Organizing Maps are an unsupervised type of neural networks that learn on their own through unsupervised competitive learning by mapping the weights of the nodes to conform to the input data presented to the network [17]. SOMs are represented using low dimensional (usually 2- D) representation of the input data. It has only one layer in which each node also called the neuron has 2 properties: its position in the map (x,y coordinates) and its codebook (CB) vector. The CB vectors for neurons have the same dimensions as the input data (normally 1 x m for m-dimensional data space). The training in SOM consists of 3 distinct phases:

*1) Initialization:* SOM can be initialized in a random or linear manner. In random initialization, each codebook vector is assigned a random value for the dimension representing a particular attribute. Linear initialization chooses the codebook vector in the subspace formed by the eigenvectors for the two greatest eigenvalues.

*2) Rough-Training:* The first phase of SOM training has a higher neighborhood radius and learning rate with a fewer number of epochs. This phase is also called fast learning because the CB vectors for neurons update significantly based on the proximity to the best matching unit (BMU).

*3) Fine-Training:* It consists of a larger number of epochs, small learning rate and smaller neighborhood width. The fine training starts with a smaller radius and the CB vectors change to a smaller extent as compared to coarse learning.

The SOM algorithm applied in this work is summarized as Algorithm 3.

**Algorithm 3** SOM algorithm

1: Randomly initialize the codebook vectors for each neuron.
2: **Repeat**
3: Input faults data to the network in a random sequential manner.
4: Identify the BMU through minimum $L^2$ distance from the input vector.
5: Calculate the neighborhood radius that starts from the initial SOM radius and decreases exponentially with each epoch as $R_n(\sigma(t)) = \sigma_o e^{-t/\lambda}$ where $\sigma_o$ is the initial SOM radius and $\lambda$ is the time constant given by the ratio of total epochs and map radius.
6: Adjust the weights of the nodes to resemble the input vector such that the nodes in close vicinity to the BMU experience higher change in their weights, i.e. $W(t+1) = W(t) + \Omega(t)L(t)(I(t) - W(t))$, where I(t) and W(t) are input and CB vectors, $L(t) = L_o e^{-t/\lambda}$ and $\Omega(t) = e^{-(L^2)^2/2\sigma^2(t)}$. Here $L_o$ and $L^2$ are the initial SOM learning rate and Euclidean distance respectively.
7: **Until** error parameters are minimized.

### D. Local Outlier Factor (LOF)

This is a density based local outlier analysis algorithm proposed for outlier detection in data sets [18]. The detection is based on cluster density in the surroundings of each data point and the factor can be represented as a continuous value with higher values indicating the data point being away from a dense cluster. The parameter that affects the performance of the algorithm is MinPts that is the number of data points defining the neighborhood of an object. The LOF calculation algorithm is summarized in Algorithm 4:

**Algorithm 4** LOF algorithm

1: Calculate the reachability distance of an object $p$ with respect to another object $o$ as: $Rd_k(p,o) = max\{d_k(o), L^2(p,o)\}$ where $d_k(o)$ defines the k-distance of the object $o$.
2: Calculate the local reachability distance (lrd) that is the inverse of the average reachability distance based on the Min-pts neighborhood and given for an object p by: $lrd_{MinPts}(p) = [(\sum_{o\varepsilon MinPts(p)} Rd_{MinPts}(p,o))/N_{MinPts}(p)]^{-1}$.
3: Calculate the LOF score that is defined as the averaged ratio of local reachability factor of $p$ and the local reachability factor of its MinPts neighbors. Mathematically, it is expressed as $LOF_{Minpts}(p) = \sum_{o\varepsilon MinPts(p)} \frac{lrd_{MinPts}(o)}{lrd_{MinPts}(p)} . N_{MinPts}(p)^{-1}$

### E. Local Outlier Probabilities (LoOP)

This is an enhancement to the LOF algorithm by assigning an outlier score (LoOP) in the range of [0,1] [19]. The LoOP values show the degree to which an object in a cluster can be identified as an outlier. The LoOP results from this technique outperforms the LOF scores as they have a fixed range so the degree of outlierness can be compared for different data distributions with relative ease. The algorithm for calculating LoOP values is summarized below:

**Algorithm 5** LoOP algorithm

1: Calculate the probabilistic set distance of an object $o$ in a data set S with significance $\mu$ as $Pd(\mu, o, S) = \mu.\eta(o, S)$ where $\eta$ denotes the standard deviation of the object with regards to the $L_2$ distance.
2: Calculate the probabilistic local outlier factor (PLOF) that is the ratio of object densities around the object and the expected value of the densities around all objects in the data set and expressed as: $PLOF_{\mu,s}(o) = \frac{Pd(\mu,o,S(o))}{E_{s\varepsilon S(o)}[Pd(\mu,o,S(s)]} - 1$.
3: Normalize the PLOF values and convert into probability values (LoOP) by: $LoOP_s(o) = max\{0, erf(\frac{PLOF_{\mu,s}(o)}{nPLOF.\sqrt{2}})\}$, where $nPLOF = \mu.\sqrt{E[(PLOF)^2]}$.

## IV. Data Attributes & Preprocessing

Data filtering and preprocessing are few of the most crucial processes in the knowledge discovery process before applying any clustering algorithms. The selected data attributes in our analysis have different types, distributions and ranges. A summary of the data attributes' features are presented in Table I.

Along with distinct data types and ranges, the attributes have different distribution properties. For instance, the Date attribute has somewhat uniform distribution whereas the Lead Time has a distribution curve similar to Gamma distribution. The categorical data attributes are assigned numerical values based on the proximity of geographical regions in case of Region attribute and similarity in root cause analysis (RCA) for Fault Cause attribute. Since the data range is not uniform for the attributes, we perform the standard / z-score normalization to convert the data range on a uniform scale for all attributes. As an example, the data range reduction after normalization for Lead Time attribute is given in Fig. 1.

## V. Learning Results and Discussion

### A. K-means, Fuzzy C-means

We use the elbow method to determine optimal number of clusters for our K-means and FCM analysis. For both the algorithms, the optimal value of $K$ after which further clustering gives linear improvement is 5. Several iterations of each algorithm are run to determine the performance in terms of minimum SSE. The SSE for the entire data with $K$ clusters each with centroid $c_i$ is calculated using

$$SSE = \sum_{i=1}^{K} \sum_{x\varepsilon C_i} L^2(c_i, x) \quad (3)$$

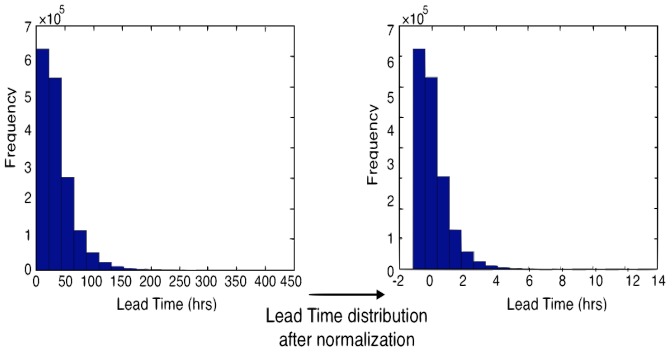


Fig. 1. Range variation in Lead Time scale after z-score normalization

where $L^2(c_i, x)$ denotes the Euclidean distance between the centroid and a data point $x$. The optimal SSE results for each algorithm are picked and presented in Fig. 2. The K-means clustering exhibits better SSE results based on which we select it as the technique used for extracting insights from the raw NFL data. The FCM does not perform optimally for our data set because of overlap within data attributes between multiple clusters. FCM algorithm terminates within fewer iterations for this kind of big data but K-means creates a larger separation between clusters which is desired for distinguishing distinct features in each cluster.

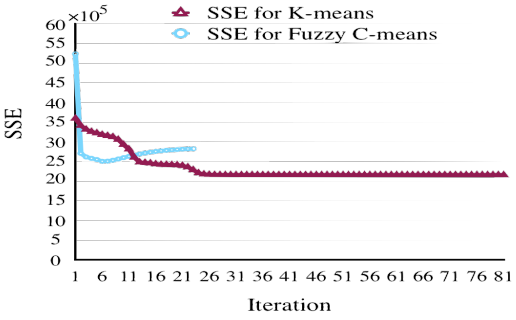


Fig. 2. Optimal SSE results for K-means, FCM

After establishing that K-means clustering outperforms FCM with better SSE results, we investigate the clusters formed with respect to the Lead Time attribute as the most critical key performance indicator (KPI) for customer satisfaction. The Lead Time distribution for each cluster is presented in Fig. 3 where $\mu$ and $\sigma$ denote the mean and standard deviation. It is observed that cluster 5 has the worst resolution times with highest mean Lead Time. We investigate other associated attributes in cluster 5 to find linkages with the critical KPI and find that the Region attribute has mean 2.30 and constitutes NFL data originating mostly from region 1 of the service area, the Fault Type has a major contribution from sync loss and slow browsing due to network parameters, the Fault Date is uniformly distributed which indicates this attribute has no distinction in this cluster and the critical Fault Time has highest percentage distribution between 1

TABLE I
DATA ATTRIBUTES PROPERTIES

| Attribute | Type | Range | Description |
|---|---|---|---|
| Date | numerical, circular and continuous | (1,13] | reflects the month and day of fault occurrence |
| Region | categorical, linear and discrete | [1,7] | five regions assigned values based on geographical separation |
| Time | numerical, circular and continuous | [0,24) | fault occurrence time in hh:mm |
| Fault Cause | categorical, circular and discrete | [1,100] | 92 different fault causes assigned numerical values based on similarities in root cause |
| Lead Time | numerical, linear and discrete | [0,437) | fault resolution time in hours |

pm and 4 pm. These insights show that there are certain geographical regions, fault types and critical service times for the operator. To enhance overall customer experience, reduce service delays and therefore increase customer retention, the operator must improve the network and performance standards in region 1, perform further root cause analysis to avoid the highlighted fault causes and ensure downtimes reduction during the mentioned service critical times.

### B. Self-Organizing Maps

Due to its high complexity and large data set size, we train our SOM for a 15x15 network grid with codebook vectors of dimension 1x5 for each node / neuron. The first step in SOM learning is choosing the initial network parameters such as neighborhood radius and learning rates for both coarse and fine learning phases. The coarse phase starts with the following parameters: initial map radius $\sigma_o$= 10, time constant $\lambda$= Number of epochs / map radius = 200/10 = 20 and neighborhood radius $\sigma = \sigma_o e^{-t/\lambda} = 10e^{-t/20}$. The radius reduction implies that the coarse learning phase, i.e. when neighborhood radius $> 1$, completes at 55 epochs. We notice that overfitting causes increase in error parameters during fine training. To avoid overfitting, we keep the training radius above 1 for the entire fine learning phase. We conduct multiple experiments to determine the optimal initial parameters and observe that when $\sigma_o = 5$ and $L_o = 0.1$, the error parameters topological error and DBI show least values of 3.97% and 16.93 respectively at the 154th epoch after which overfitting increases topological error value (Fig. 4). Topological error is the percentage of data points in an epoch for which the 1st and 2nd best matching nodes are not neighbors. DBI is also a measure for evaluating clustering algorithms as it is an average ratio of intra cluster variance and inter cluster distance for all the clusters. Lower DBI values indicate better clustering and higher separation between cluster centroids.

We train the SOM network with the optimal initial parameters for 154 epochs and obtain a smooth distribution for all the dimensions, each dimension representing the CB vectors for the respective attributes in a 2-D 15x15 grid. The data distribution shows highest percentage of data points in neurons with coordinates (15,15), (15,1) and (15,4) that have lead times between 29-31 hours which according to the input data is close to the mean lead time and should constitute the majority of data. We analyze the performance critical KPI Lead Time's distribution over the SOM and observe that the pain point neurons with highest fault resolution times reside near the origin and bottom corner left in the SOM (Fig. 5). To analyze

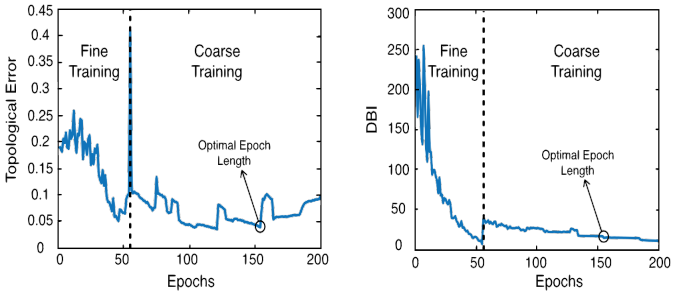


Fig. 4. Error metrics for optimal initial parameters

the association of other data attributes with nodes exhibiting high Lead Time values, we plot the SOM distribution for each attribute separately (Fig. 6 - due to space constraint, we only plot the color gradient of attribute distribution). The trained SOM network gives a well separated distribution for each attribute as seen from Figs. 5 and 6. From the circled nodes in Figs. 5 and 6, we infer that high lead times mostly occur during the months of May – July, the geographical locations associated with these long duration outages originate mostly in regions 1 and 2, the critical fault occurrence times are between 12-2 pm and 8-9 pm, and the associated fault cause numbers indicate Sync Loss and Browsing issues as the core reasons for delayed fault resolutions in these regions. The SOM network thus provides similar insights as K-means analysis to the service provider in terms of the spatio-temporal pain points in the network.

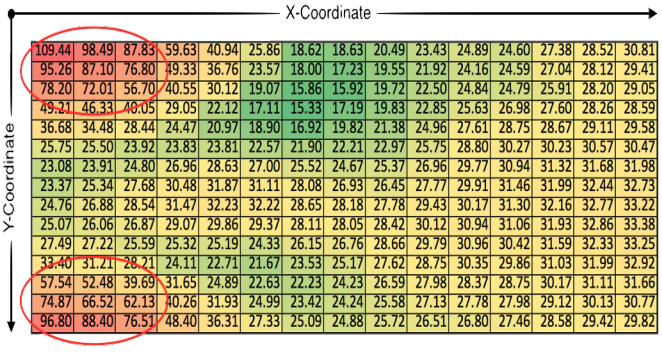


Fig. 5. Lead Time distribution in the SOM network

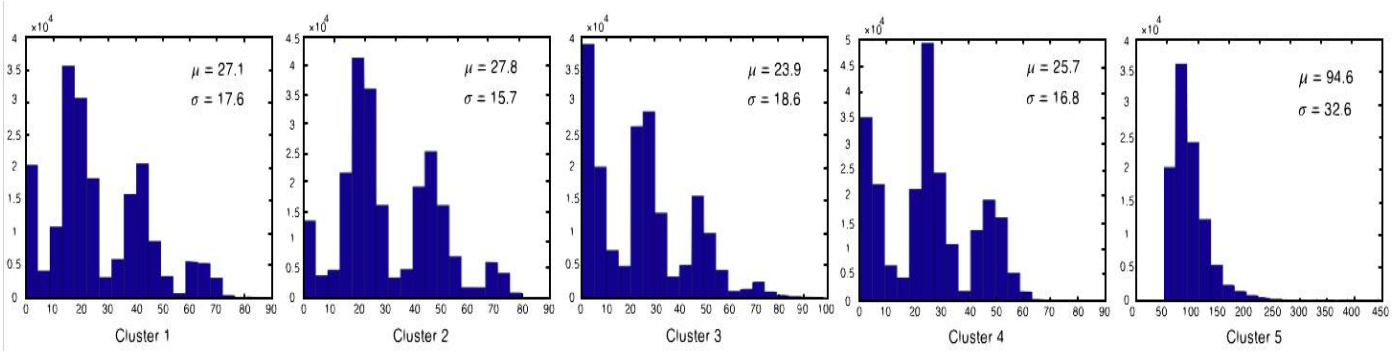


Fig. 3. Lead Time attribute for K-means clusters

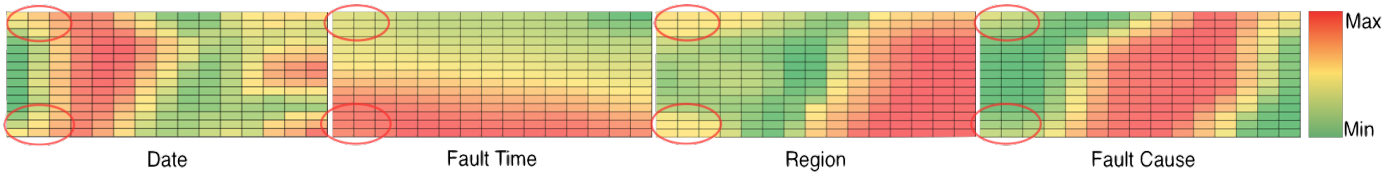


Fig. 6. Date, Fault Time, Region and Fault Cause distributions in the SOM network

### C. SOM with density based anomaly detection

We apply different density based local outlier determination algorithms (LOF, LoOP) to consolidate on the trained SOM network and determine the degree to which every node exhibits anomalous behavior. The outlier results after applying these algorithms are given in Figs. 7 and 8. For LOF, we increment MinPts in multiples of 10 and observe that the maximum LOF values are obtained at MinPts = 20. From Fig. 7, the nodes with higher anomaly factor (circled in red) using LOF algorithm tend to be located at the corner nodes in the SOM. The LoOP analysis (we set $\mu$=3) normalizes the anomaly factor and provides a clear distinction of the anomalous nodes from the rest of the network (Fig. 8). This is because LoOP algorithm is independent of MinPts and gives a relative measure of outlierness. The attributes for anomalous neurons include extremely early morning fault times (2-5 am), geographical locations based in regions 2 and 3, occurrence dates within the months of June - July and resolution times around the mean value (about 30 hours). Although the probability of occurrence of the anomalous scenarios is low, the operator must be proactive in avoiding faults in the highlighted spatio-temporal regions.

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1.76 | 1.50 | 1.40 | 1.17 | 1.13 | 1.05 | 1.16 | 1.23 | 1.10 | 0.99 | 1.24 | 1.18 | 1.73 | 2.05 | 2.38 |
| 1.51 | 1.28 | 1.15 | 1.06 | 1.06 | 1.00 | 1.00 | 1.04 | 1.02 | 0.92 | 1.00 | 1.11 | 1.43 | 1.44 | 1.56 |
| 1.20 | 1.10 | 0.93 | 0.98 | 0.95 | 1.00 | 0.90 | 0.91 | 0.89 | 0.88 | 0.93 | 1.10 | 1.03 | 1.19 | 1.21 |
| 1.13 | 1.00 | 0.95 | 0.93 | 0.92 | 0.98 | 0.97 | 0.98 | 0.96 | 0.94 | 1.00 | 1.10 | 1.04 | 1.07 | 1.18 |
| 1.18 | 1.03 | 0.95 | 0.99 | 1.02 | 1.08 | 1.02 | 1.03 | 0.96 | 0.99 | 1.03 | 1.04 | 0.94 | 1.07 | 1.18 |
| 1.26 | 1.11 | 0.90 | 0.95 | 0.96 | 1.00 | 1.05 | 1.01 | 1.00 | 1.02 | 1.04 | 1.06 | 0.88 | 0.98 | 1.18 |
| 1.19 | 0.99 | 0.87 | 0.95 | 1.06 | 1.01 | 1.00 | 0.99 | 1.00 | 1.01 | 1.01 | 1.01 | 0.92 | 0.97 | 1.14 |
| 1.23 | 1.01 | 0.90 | 0.96 | 1.07 | 1.01 | 0.98 | 0.98 | 1.03 | 0.98 | 0.98 | 1.05 | 0.97 | 0.99 | 1.14 |
| 1.15 | 1.05 | 0.94 | 1.07 | 1.06 | 1.01 | 0.99 | 0.99 | 1.00 | 0.99 | 0.99 | 1.10 | 1.00 | 0.99 | 1.08 |
| 1.23 | 0.98 | 0.89 | 1.06 | 1.07 | 0.99 | 1.03 | 1.02 | 1.01 | 0.97 | 1.01 | 1.09 | 0.93 | 0.98 | 1.08 |
| 1.22 | 1.02 | 0.93 | 0.96 | 1.02 | 1.01 | 1.00 | 1.00 | 1.04 | 0.97 | 1.09 | 0.99 | 0.97 | 1.07 | 1.12 |
| 1.18 | 0.97 | 0.89 | 0.97 | 0.99 | 0.94 | 0.98 | 0.99 | 0.98 | 0.97 | 1.07 | 0.93 | 0.90 | 1.09 | 1.13 |
| 1.08 | 0.91 | 0.91 | 1.06 | 1.01 | 0.94 | 0.95 | 0.82 | 0.93 | 0.95 | 0.96 | 0.80 | 0.88 | 0.98 | 0.99 |
| 1.26 | 1.08 | 1.01 | 0.98 | 0.98 | 0.99 | 1.07 | 1.05 | 0.98 | 1.12 | 1.04 | 0.99 | 1.02 | 1.08 | 1.06 |
| 1.67 | 1.53 | 1.29 | 1.11 | 1.00 | 1.09 | 1.18 | 1.06 | 1.08 | 1.19 | 1.13 | 1.07 | 1.25 | 1.22 | 1.34 |

Fig. 7. LOF (Local Outlier Factor) values for MinPts=20

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0.27 | 0.20 | 0.20 | 0.22 | 0.23 | 0.23 | 0.24 | 0.21 | 0.22 | 0.28 | 0.48 | 0.62 | 0.95 | 0.99 | 1.00 |
| 0.09 | 0.04 | 0.04 | 0.10 | 0.16 | 0.18 | 0.17 | 0.14 | 0.16 | 0.23 | 0.38 | 0.50 | 0.89 | 0.91 | 0.94 |
| 0 | 0 | 0 | 0 | 0.07 | 0.10 | 0.06 | 0.04 | 0.05 | 0.14 | 0.25 | 0.37 | 0.45 | 0.76 | 0.81 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.08 | 0.20 | 0.27 | 0.30 | 0.33 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.06 | 0.14 | 0.18 | 0.22 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.03 | 0.10 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0.08 | 0.02 | 0.01 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0.21 | 0.17 | 0.16 | 0.15 | 0.078 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0.23 | 0.21 | 0.24 | 0.27 | 0.23 | 0.17 | 0.13 | 0.08 | 0.04 | 0.05 | 0.12 | 0.06 | 0 | 0 | 0 |
| 0.28 | 0.26 | 0.27 | 0.33 | 0.31 | 0.28 | 0.26 | 0.24 | 0.21 | 0.23 | 0.25 | 0.24 | 0.13 | 0.07 | 0.02 |
| 0.37 | 0.34 | 0.33 | 0.37 | 0.35 | 0.38 | 0.36 | 0.34 | 0.32 | 0.37 | 0.38 | 0.34 | 0.25 | 0.19 | 0.19 |

Fig. 8. LoOP (Local Outlier Probabilities) values for $\mu$ =3

### D. Clustering efficiency analysis

Finally after analyzing insights from our K-means clustering and SOM results, we compare the clustering efficiency of K-means, FCM and SOM. Since each node in the SOM can be considered as an independent cluster, we apply K-means clustering on the trained SOM network with K = 5 to obtain uniform number of clusters for each technique. The clustering performance is evaluated in terms of SSE and DBI values, the results of which are summarized in Table II. The large variation in SSE value for SOM as compared to K-means and FCM is because we have 225 SOM nodes which are significantly less than the total number of data points clustered using K-means and FCM. However, the DBI result shows that SOM outperforms K-means and FCM algorithms. Lower DBI indicates densely populated and well separated clusters rendering the insights extracted from SOM more reliable.

For the hypotheses stated earlier, we can summarize the

TABLE II
CLUSTERING EFFICIENCY COMPARISON

| Method | Number of iterations | SSE | DBI |
|---|---|---|---|
| K-means | 81 | 2156788.2 | 15.68 |
| Fuzzy C-means | 23 | 2822823.5 | 17.98 |
| Clustered SOM | 48 | 1136.6 | 12.89 |

following as findings of our analysis:

• Both the K-means and SOM leverage unsupervised clustering to identify spatio-temporal patterns linked with high fault lead times.

• Clustered SOM yields lower error metrics; hence it produces clustering results that have a higher credibility.

• LoOP applied on SOM results in efficient and more focused anomaly detection in NFL data.

• The analysis of clustered data reveals that that highest fault resolution lead times are attributed to the summer months (May-July) and have a higher occurrence probability in regions 1 and 2. Most of these faults are caused by "Sync Loss" and "Slow browsing" issues.

• To enable efficient proactive self-healing mechanisms in future, the network operator must devise a SON engine that leverages insights from applying the highlighted data mining techniques on the continuously produced NFL.

## VI. CONCLUSIONS

In this paper, we have used different data mining techniques for extracting critical network pain points and anomalies from the CRM based broadband network failure log database of a nationwide operator. We have considered multiple unsupervised clustering techniques and density based local outlier detection algorithms as tools for our analysis. Our results indicate that SOM outperforms K-means and Fuzzy C-means when clustering the complaints in the NFL dataset with lower DBI value. The anomaly detection results show that LoOP values are more reliable in detecting the anomalous nodes in the SOM network due to their independence of the MinPts factor.

We have also analyzed the SOM and K-means clustered data based on the Lead Time KPI. The insights from the clustering analysis highlight the dates in the calendar year, geographical locations, critical times of the day and severe fault causes that are likely to be associated with longer outages. Similarly, the anomaly detection algorithms identify the spatio-temporal signatures of the rarely occurring network faults. To enhance customer retention and improve QoS, the operator should adapt a proactive self-healing strategy to minimize customer complaints and network outages in the highlighted critical spatio-temporal regions.

## ACKNOWLEDGMENT

This work is supported in part by the National Science Foundation under Grant No. NSF-CNS-1619346. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

## REFERENCES

[1] CISCO, "Cisco visual networking index: Global mobile data traffic forecast update, 2015 to 2020 white paper," 2016.

[2] Cartesian, "The emergence of LTE small cells and 5G," 2015. [Online]. Available: http://www.cartesian.com/the-emergence-of-lte-small-cells-and-5g/

[3] A. Imran, A. Zoha, and A. Abu-Dayya, "Challenges in 5G: how to empower SON with big data for enabling 5G," *IEEE Network*, vol. 28, no. 6, pp. 27–33, Nov 2014.

[4] R. Becker, R. Caceres, K. Hanson, J. M. Loh, S. Urbanek, A. Varshavsky, and C. Volinsky, "Clustering Anonymized Mobile Call Detail Records to Find Usage Groups," *The First Workshop on Pervasive Urban Applications (PURBA)*, 2011.

[5] A. Bascacov, C. Cernazanu, and M. Marcu, "Using data mining for mobile communication clustering and characterization," in *Applied Computational Intelligence and Informatics (SACI), 2013 IEEE 8th International Symposium on*, May 2013, pp. 41–46.

[6] T. Vafeiadis, K. I. Diamantaras, G. Sarigiannidis, and K. C. Chatzisavvas, "A comparison of machine learning techniques for customer churn prediction," *Simulation Modelling Practice and Theory*, vol. 55, pp. 1–9, 2015.

[7] S. K. I. Hasitha Indika Arumawadu, R. M. Kapila Tharanga Rathnayaka, "Mining Profitability of Telecommunication Customers Using K-Means Clustering," *Journal of Data Analysis and Information Processing*, vol. 3, pp. 63–71, 2015.

[8] M. Horváth and A. Michalkova, "Monitoring Customer Satisfaction in Service Industry: A Cluster Analysis Approach," *Quality Innovation Prosperity*, vol. 16, no. 1, 2012. [Online]. Available: http://EconPapers.repec.org/RePEc:tuk:qipqip:v:16:y:2012:i:1:5

[9] T. Velmurugan, "Performance based analysis between k-Means and Fuzzy C-Means clustering algorithms for connection oriented telecommunication data," *Applied Soft Computing*, vol. 19, pp. 134–146, 2014.

[10] F. Bação, V. Lobo, and M. Painho, "Self-organizing Maps as Substitutes for K-Means Clustering," in *Proceedings of the 5th International Conference on Computational Science - Volume Part III*, ser. ICCS'05. Berlin, Heidelberg: Springer-Verlag, 2005, pp. 476–483.

[11] U. A. Kumar and Y. Dhamija, "Comparative analysis of SOM neural network with K-means clustering algorithm," in *Management of Innovation and Technology (ICMIT), 2010 IEEE International Conference on*, June 2010, pp. 55–59.

[12] H. Farooq, A. Imran, and A. Abu-Dayya, "A multi-objective performance modelling framework for enabling self-optimisation of cellular network topology and configurations." *Trans. Emerging Telecommunications Technologies*, vol. 27, no. 7, pp. 1000–1015, 2016. [Online]. Available: http://dblp.uni-trier.de/db/journals/ett/ett27.htmlFarooqIA16

[13] A. Zoha, A. Saeed, A. Imran, M. A. Imran, and A. Abu-Dayya, "A learning-based approach for autonomous outage detection and coverage optimization." *Trans. Emerging Telecommunications Technologies*, vol. 27, no. 3, pp. 439–450, 2016. [Online]. Available: http://dblp.uni-trier.de/db/journals/ett/ett27.htmlZohaSIIA16

[14] O. Onireti, A. Zoha, J. Moysen, A. Imran, L. Giupponi, M. A. Imran, and A. Abu-Dayya, "A Cell Outage Management Framework for Dense Heterogeneous Networks," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 4, pp. 2097–2113, April 2016.

[15] A. Zoha, A. Saeed, A. Imran, M. A. Imran, and A. Abu-Dayya, "Data-driven analytics for automated cell outage detection in Self-Organizing Networks," in *Design of Reliable Communication Networks (DRCN), 2015 11th International Conference on the*, March 2015, pp. 203–210.

[16] P.-N. Tan, M. Steinbach, and V. Kumar, *Introduction to Data Mining*. Pearson Addison Wesley, 2006.

[17] S. M. Guthikonda, *Kohonen Self-Organizing Maps*. Wittenberg University, December 2005.

[18] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "LOF: Identifying Density-based Local Outliers," *SIGMOD Rec.*, vol. 29, no. 2, pp. 93–104, May 2000.

[19] H.-P. Kriegel, P. Kröger, E. Schubert, and A. Zimek, "LoOP: Local Outlier Probabilities," in *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, ser. CIKM '09. New York, NY, USA: ACM, 2009, pp. 1649–1652.