Quantifying the Service Performance Impact of Self-Organizing Network Actions

Swati Roy[△], David Applegate[‡], Zihui Ge[‡], Ajay Mahimkar[‡], Shomik Pathak[‡], Sarat Puthenpura[‡]

Princeton University [△] AT&T [‡]

***Abstract*—As smartphone users increasingly rely on cellular networks to access voice, video, and web applications, guaranteeing good performance and high availability is more important than ever. Historically, managing cellular network configuration has been a manual, error-prone process; recently, automated solutions such as SON (Self-Organizing Networks) controllers are being deployed for dynamic tuning of network configuration to improve end-user service performance under dynamic network and traffic conditions. SON automates many aspects of cellular network configuration, but it is nonetheless susceptible to software bugs and expected traffic changes that could result in sub-optimal performance. In this paper, we propose a capability (Veracity) to analyze and quantify the performance effects of SON actions. Assessing the effects of SON control is difficult because of the dynamic nature of SON and the dependency of end-user performance on factors such as radio channel quality, mobility and traffic load. Veracity addresses these using model-driven impact detection and quantification. Our evaluation using data collected from an operational cellular network demonstrates that Veracity is accurate. Veracity is now being used by the service providers' field operation teams for the assessment of SON effectiveness in arenas and stadiums.**

I. Introduction

The recent proliferation of smartphones and mobile applications have induced a dramatic increase in traffic volumes on cellular networks. Cellular service providers continuously aim to maintain excellent quality of service for millions of smart connected devices. Effectively managing and optimizing an operational cellular network is challenging because of the large number of network components, complicated network topology, rapid evolution of technologies (e.g., GSM to UMTS to LTE), overlaid circuit and packet switched architectures, different layers (macro cell towers, outdoor and indoor small cells), complex interactions between applications and network protocols, and continuous changes to the network as a result of software updates and hardware modifications.

Traditionally, network configuration management and performance optimization of cellular networks has been manual and error prone, thus significantly increasing operational expenditure (OPEX). For example, a human operator would need to go through a sequence of esoteric configuration parameter settings in order to obtain an optimal service performance using available radio network resources, often by trial and error. Automating network configuration management and performance optimization tasks can help reduce operational cost, errors, and downtime.

978-3-901882-85-2

SON (Self-Organizing Networks) [2], [3], [18] is a technology that promises to make management, configuration and optimization of a large operational cellular networks simpler, faster and automatic to improve end-user service performance over continuously changing network and traffic conditions. With the deployment of SON manual errors are eliminated but like any software, it may have bugs and might degrade service performance.

Why a classic pre/post or before/after impact assessment will not work for SON? For impact assessment of SON actions, one could apply classical change detection techniques to compare the performance before and after the SON action, and identify if the changes are statistically significant. However, a simple pre/post or before/after change detector is not sufficient because of the unique characteristics of SON mentioned below:

1. **Dynamic nature of SON.** The iterative, rapid configuration tuning by SON creates a highly dynamic network environment that makes it difficult to assess the effects of configuration changes that the SON controller introduces. Multiple SON actions from possibly multiple SON applications that occur in succession and on the same set of cell towers make it difficult to understand the effects of any single action. It is important to evaluate the efficiency of SON tuning (e.g., tracking how many unnecessary intermediate sub-optimal changes are made). Classic time-series driven impact analysis [4], [13]–[15], [19], [26] does not apply in such a dynamic setup because it is difficult to establish a baseline in such a situation.
2. **Expected performance impact of SON.** SON controllers can induce improvement in one performance metric at the cost of a minor degradation in another. It is important to capture this behavior when analyzing the overall performance impact of SON. For example, up-tilting an antenna leads to an increase in the number of end-users served by the cell tower but that could result in a minor degradation in average data throughput experienced by all the end-users. If one is not aware of this expected behavior, performing just pre/post time-series analysis would lead to wrong assessment on the overall service performance impact.
3. **Unpredictable external network events.** Equipment failures, network congestion or changes in radio channel quality (due to introduction of an interfering source) are unpredictable and can significantly impact end-users

service performance. Overlapping SON actions on a cell tower and these network events at neighbors can make impact quantification difficult. Hence, it is important to capture such behaviors when analyzing the performance impact of SON changes.

4. **SON related factors.** Incorrect SON parameters, SON software bugs and erroneous implementation of SON algorithms might degrade service performance. In other words, SON is just a tool which might yield a sub-optimal performance under certain network and traffic conditions.

Veracity Approach: In this paper, we present Veracity, a new model-driven change impact analysis approach to accurately quantify the service performance impact of automated SON actions. We carefully capture the traffic Load, user Mobility, and Radio environment (LMR) metrics that can change behavior depending on the SON action and has the potential to induce an impact on service performance. Veracity first builds a statistical dependency model using piece-wise linear regression between historical Service Quality Metrics (SQMs) and a list of dependent metrics (LMR) - explained in section III-A. The resulting *model coefficients* and the *instantaneous* LMR metrics after SON actions are used to compute the *estimated* SQMs values. By comparing the *estimated* SQMs with the *instantaneous* SQMs observed after the SON actions, Veracity accurately quantifies the statistical service performance impact.

For example, if the SON actions result in an increase in instantaneous traffic, then the dependency model would accurately compute the expected change in the SQMs. If the *instantaneous* SQM observed after the SON actions match the *estimated* values, then Veracity labels this case as no impact. However, if the *observed* SQM is statistically higher than the *estimated* SQM, Veracity labels this scenario as a performance improvement; otherwise, Veracity labels the scenario as a performance degradation. Thus, Veracity does not require an explicit comparison of time-series before and after the SON actions. Instead, Veracity leverages the dependency model between the SQMs and the LMR metrics to accurately quantify the performance impacts.

The novelty of the paper does not lie in regression but in identifying LMR metrics that are indicative of SQMs so that one is aware of the expected change in the SQMs after the SON action is implemented. We apply regression as a tool and the intuition behind using supervised learning, especially regression approach, is based on the observations made using data collected from operational networks. We experimented with various regression algorithms before deciding upon piece-wise linear regression. We couldn't apply non-linear regression algorithms like polynomial regression [24] because it is a hard problem to determine which functional model to use and piece-wise linear regression is flexible and reasonably generic. We also present the comparison of the chosen piece-wise linear regression over simple linear regression in section IV-A.

Our Contributions:

1. We present Veracity, a model-driven change impact analysis approach that tackles the dynamic nature of SON, captures the expected impacts induced by it and adds robustness to handle the unpredictable external network events with zero false positives.
2. We thoroughly evaluate Veracity using real-world data collected from operational cellular networks[1] in section IV-A and demonstrate its effectiveness over existing time-series pre/post approaches like Litmus [14] used by service providers' operation teams in section IV-B. We also show that the piece-wise linear regression is better suited to capture the dependency model and has an improved error as compared to linear ridge regression [8] in section IV.
3. Veracity is now being used by the service providers' field operation teams for quantifying the effectiveness of SON in arenas and stadiums. We share our operational experiences using three case studies in Section V.

II. Background

In this section, we provide a brief background on the cellular network architecture in Section II-A, Self-Organizing Network (SON) and one use case application in Section II-B.

A. Cellular Networks

The UMTS and LTE cellular networks comprise of a radio access network (RAN) and a core network (CN). The smartphones use the air interface for connecting to the cellular RAN network base stations (referred to as NodeB in UMTS and eNodeB in LTE). Each base station can transmit using multiple overlaid standardized frequency blocks, each having a center called a carrier frequency. The UMTS network supports voice services over circuit switched core network and data services over packet switched core network. LTE offers voice as well as data over its packet switched network. The User Equipment (UE) sets up an end to end connection/channel with the help of the radio and core networks nodes in order to be able to start voice/data services. The cellular service provider periodically collects a wide variety of service and network performance measurement metrics from the base stations.

B. Self-Organizing Network (SON)

Self-Organizing Networks (SON) [2], [3], [18] uses the LTE and UMTS metrics to configure, manage and optimize the operational cellular networks automatically. The configuration changes executed by the SON controllers should result in an overall improvement of end-user's service performance. For example, let us assume that 5 users receive service from a cell tower (with an average data throughput of 10 Mbps) and 5 users are out of coverage because of neighboring cell tower failure or coverage hole. In such a scenario, SON would execute configuration changes such as up-tilting the antenna to cover the 5 users that were out of service. Because of increased usage on the cell tower, the nearby users that previously received 10 Mbps, now receive only 8 Mbps data throughput. The far-away users individually receive a data throughput of 4 Mbps. Thus, we observe that after the antenna tilt change, the number of users increase from 5 to 10; however, the average

[1]To protect proprietary data, we explicitly do not show any service performance numbers.

Service Quality Metrics	Explanation
Accessibility	Ratio between successful call establishment over all call attempts.
Retainability	Ratio between successful call termination over successful call establishment.
Downlink Throughput	Total number of bits received per second.

TABLE I
SERVICE QUALITY METRICS AND THEIR EXPLANATION.

data throughput (a user perceived service quality metric) goes down from 10 Mbps to 6 Mbps. This is an expected impact in throughput because of the increase in the number of users being served. If one only focuses on throughput (classic pre-post), then it would result in wrong assessment of the SON action; however, the overall service impact of the SON action is good because the increase in the number of users served comes at an acceptable cost of minor degradation in throughput.

III. Design and Implementation

We now describe SQMs and LMR metrics and our approach of selecting LMR metrics which are indicative of SQMs. We then describe how we relate LMR metrics to accurately estimate SQMs.

A. SQM and LMR

Service Quality Metrics (SQMs) [1] capture the quality of service performance experienced by end-users. Table I provides a description of three main service quality metrics used in the paper: (i) *Accessibility* shows the ratio between successful call establishment over all call attempts in the defined time window, (ii) *Retainability* shows the ratio between successful call termination over successful call establishment in the defined time window, and (iii) *Downlink throughput* is a measure of bits per second delivered to the end-users.

We need every metric that could help us in identifying the behavior of SQMs. We categorize network and traffic measurement metrics collected from the base station that likely impact the SQMs into **Traffic Load, Mobility and Radio Environment (LMR)**. Tables II provides a summary of LMR categories: (i) *Traffic* includes number of sessions, session volume and resource utilization at cell tower, (ii) *Mobility* is captured via handovers on the same carrier frequency, inter-carrier frequency and inter-RAT, and (iii) *Radio environment* is captured using block error rates (BLER), channel quality indicator (CQI) and received signal strength indicator (RSSI). Each category consists of multiple metrics such as radio environment category consists of three metrics such as RSSI, BLER and CQI. We start our experimentation with all the measurement metrics (features) extracted from the data collected and eliminate redundant features using domain knowledge to remove multi-collinearity. Finally, based on the inputs from the radio network experts, we choose those LMR metrics from the filtered list that can effectively capture traffic patterns, mobility and radio environment of the network.

One could argue that a single metric is sufficient to effectively estimate the SQMs and quantify performance impact. Hence, we now demonstrate the need for multiple LMR metrics as opposed to a single metric, for capturing the

Traffic Parameters	
RRC (Successful Radio Resource Connection)	User is successfully allocated some radio resources to send or receive data.
DL PDCP Volume (Downlink Packet Data Convergence protocol)	Traffic volume in the downlink direction.
UL PDCP Volume (Uplink Packet Data Convergence protocol)	Traffic volume in the uplink direction.
DL PRB utilization (Downlink Physical Resource Block)	Total number of physical resource blocks utilized in the downlink direction.
UL PRB utilization (Uplink Physical Resource Block)	Total number of physical resource blocks utilized in the uplink direction.
Handovers Parameters	
IRAT (Inter Radio Access Technology) Redirect	Number of successful redirection attempts from LTE to UMTS (cross technology).
Intra frequency Handovers Attempts	Total number of handover attempts within the same carrier frequencies.
Inter frequency Handover Attempts	Total number of handover attempts between carrier frequencies.
Radio Environment Parameters	
RSSI (Relative Signal Strength indicator)	Total signal strength received at the (e)NodeB.
BLER (Block error rate)	Total percentage of user data blocks received in error at the cell tower. This metric captures bad coverage.
CQI (Channel quality indicator)	Quality of the channel as reported by the user equipment and captures downlink interference.

TABLE II
TRAFFIC LOAD, MOBILITY AND RADIO ENVIRONMENT METRICS AND THEIR EXPLANATION.

relationship with SQMs. These multi-variate dependencies do exhibit in operational settings and play an important role for explaining the changes in the SQMs. We use a month-long data collected from 643 cell towers in the LTE network to demonstrate the result of our analysis. All metrics are aggregated every hour (one measurement per cell tower per hour). We divide our month-long data into two sets: with first 15 days of data into set A and rest in set B. Here, our aim is to find anomalies in set B with respect to set A. In order to handle the time of the day effect, we construct a unique time series, X, by combining the same hour of every day for each hour of the day, for set A (X^{setA}) and set B (X^{setB}). We use a simple median-based anomaly detector to learn a robust median and a median absolute deviation of X^{setA} for each SQM and LMR metric. In order to capture anomalies, we apply the equation 1 to calculate an anomaly score for each hour in set B and for each cell tower. We compare the anomaly score to the standard statistical threshold for deviation from norm i.e. 3×1.4826 for 99% confidence intervals.

$$AnomalyScore = \frac{X^{setB} - median(X^{setA}))}{MAD(X^{setA}))} \quad (1)$$

Where X^{setB} is the set B time-series, X^{setA} is the set A time-series, median is the 50^{th} percentile of the time-series and MAD is the median absolute deviation. We choose median and MAD because of their robustness to one-off outliers.

To understand degradations in a SQM, we only consider changes in the LMR metrics when it leads to a degradation. For a given SQM in set B, we label it to be a degradation if the anomaly score is below the lower limit of the confidence interval. Then, we apply an operational threshold on each of the SQMs (degradation of more than 1% for accessibility

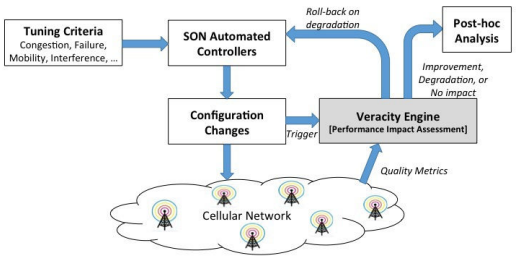


Fig. 1. SON controllers, configuration changes and impact assessment.

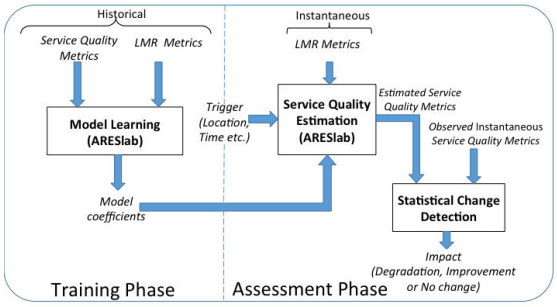


Fig. 2. Veracity design. The trigger captures the location and time information about the SON change.

and retainability, and degradation of more than 2 Mbps for downlink throughput) to capture significant degradations and eliminate subtle operationally less meaningful impacts. We observe 66 anomalous degradation points for accessibility, 32 for retainability, and 119 for downlink throughput. We then look for anomalies in all the LMR metrics. For each anomaly in the SQM, we search for a corresponding anomaly in LMR metrics. For each LMR category, we then take the union across each of its individual metrics. We observe that there are many cases where an anomaly is present in more than one of the LMR metrics for the corresponding anomaly in SQM. For example, out of 119 downlink throughput anomalies, we find 74 co-occurring anomalies in all the LMR categories, 19 in just traffic load and mobility, 6 in just traffic load and radio environment parameters, 2 in just mobility and radio environment parameters, and 13 belongs to just one of the LMR categories. For the remaining 5 cases, where none of our LMR metrics have a corresponding/co-occurring anomaly with the SQM anomaly, we manually confirm some additional configuration changes and upgrades that resulted in the service performance impacts. We observe similar results for the anomalies in retainability and accessibility. As observed from the example presented, we need each of the LMR metrics in our model learner to build a relationship between LMR metrics and every SQM.

Thus, if one is to analyze SQM individually (i.e., comparing after a SON change with before) and not taking into account the underlying LMR metrics, then the assessment could result in false positives because of lack of modeling the relationship between SQM and LMR. Our model-driven approach in Veracity carefully models the relationship and eliminates such false positives.

B. Methodology

Figure 1 shows where Veracity fits in the overall automated SON optimization, configuration tuning, and performance impact assessment system. Automated SON controllers are triggered based on the tuning criteria such as congestion, outage, interference etc. These controllers have built-in optimization and tuning algorithms that apply configuration changes to the network. Veracity is triggered by these configuration changes, and uses LMR metrics and service quality metrics for performance impact assessment. The output from Veracity is used for auditing purposes as well as a possible feedback loop to the SON controller to roll back the configuration changes when degradations are detected.

Figure 2 provides a high-level description of Veracity design. Veracity consists of two phases - (i) *model training phase* (Section III-B1) and (ii) *SQM assessment phase* (Section III-B2). In the *training phase*, Veracity builds a dependency model between LMR metrics and SQMs. It learns the underlying trends using the *historical* data. In the *assessment phase*, Veracity applies the learned *model coefficients* to a situation where SON controllers tuned network configurations, to provide the *estimated* SQMs, followed by a statistical change detection to identify and quantify the impact of SON changes.

1) Model Training Phase: Veracity applies a piece-wise linear regression model to capture the dependency between *historical* LMR metrics and SQMs. Our choice for a piece-wise linear regression model is driven by the non-linear relationships that we observed between SQMs and LMR metrics. The training interval is before the SON action was taken and could either be a single day or multiple days and continuous or discontinuous depending upon the operator's requirements. Veracity constructs a separate model for each SQM (e.g., accessibility, retainability and throughput).

Let us define an impact scope as the set of network elements (or cell towers) that can be impacted because of the actions taken by SON. For example, a tilt change by SON has the potential to impact the cell tower with the tilt change as well as its immediate neighbors. We use the impact scope to construct the time-series of LMR metrics and SQMs. Veracity then builds a model of the form given by equation 2:

$$y_j = \sum_{i=1}^{k} c_i B_i(x) \tag{2}$$

Here, c_i are the *model coefficients*, B_i are the basis functions, x are the LMR metrics and y_j is each of the SQMs. Each basis function can take any of the following forms:

1) A constant term.
2) A hinge function, which consists of breakpoints and has the form max(0, x - breakpoint) or max(0, breakpoint - x).
3) An interaction term between two or more metrics. We don't use interaction terms as it is hard to find which

functional model to use. Hence, we use additive modeling (*i.e.* no interaction terms).

We apply the ARESlab [10] implementation of multivariate adaptive regression splines (MARS) for constructing the piece-wise linear regression model. ARESlab builds the model in two-passes: (i) forward pass: adds basis functions in pairs to the model and builds an overfit model and (ii) backward pass: prunes the model to build a more generalized model and deals with the curse of multi-dimensionality. ARESlab model automatically selects LMR metrics and values of those metrics for breakpoints in the hinge functions [10].

As described in section I, our choice of MARS is based on real-world operational data, which well-suits our purpose. Adaptive splines can automatically take into account some of the non-linear relationships between certain LMR metrics and SQMs by using breakpoints. Breakpoint refers to a point where the relationship between a SQM and LMR metrics change. In other words, the relationship between LMR metrics and a SQM in the normal operating range can be quite different from the relationship under high congestion.

2) SQM Assessment Phase: Once the model is constructed using the *historical* LMR and SQM metrics (*training phase*), we use the model to compute the *estimated* SQMs for the given *instantaneous* LMR metrics (*assessment phase*) after the SON actions are implemented in the network. Finally, we quantify by computing the time-series of the differences between the *observed* and the *estimated* SQMs for the training and assessment intervals. We use rank-order tests [6], [11], [22] to compare the time-series of differences. In the absence of any SON actions, Veracity's *estimated* SQMs should closely resemble the *observed* SQMs, as described in section IV. However, in the presence of SON actions, the difference between the *estimated* SQMs and the *observed* SQMs quantifies the service performance impact. In the event of a statistical change, we check if the *observed* SQM is significantly higher than the *estimated* SQM and in this case, we conclude that SON action resulted in improvement of the service performance. On the other hand, if the *estimated* SQM is higher than the *observed* SQM, we conclude that there is a degradation of service performance. If there is no statistical change, we conclude that there is no impact on the performance. We use robust rank-order tests for detecting level changes as they are resistant to outliers. The output of the change impact analysis can be used to roll-back the SON configuration change if performance degradation is detected.

IV. Evaluation

In this section, we present the evaluation of Veracity using real-word data collected from an operational cellular network. The challenge with the evaluation using real-world data is the availability of ground truth information and the potential contamination of the data caused by external factors such as unrelated network events. We thus use a two-fold approach to conduct the evaluation. First, we carefully select a time-interval and cell towers during which there are no service performance degradations due to either network upgrades, or SON actions. This provides us with a clean data set and allows us to assess the accuracy of the model training in Veracity. In the absence of network upgrades or SON actions, the *estimated* SQMs must closely resemble the *observed* SQMs and Veracity should indicate no performance impact on SQMs. Given this scenario, we examine the effectiveness of regression in capturing the relationship between SQMs and LMR metrics. Second, we leverage the manual impact assessment of SON conducted by the field operations teams to compare the results with Veracity. This provides the ground truth information albeit on a small scale. We use two SON feature trials in the operational network. We also compare the accuracy of Veracity versus state-of-art and well established change detection approaches such as Litmus [14] and Mercury [15]. They primarily focus on the SQM metrics to analyze the impact of SON changes without taking into account the variations in LMR metrics. Litmus compares SQM metrics between a study group (where SON changes were implemented) and a control group (no SON change) and identifies the relative change in SQM on the study group. Mercury compares SQM metrics on the study group only to capture the impact of SON changes.

A. Method of capturing relationships

For this evaluation, we select cell towers that have no anomalies *i.e.* neither degradations nor improvements in the SQMs. By selecting such cell towers, we evaluate the false positives of Veracity. If Veracity indicates any impact on service quality, it will be a false positive. We also measure normalized absolute mean error for each SQM. Veracity should have minimum error irrespective of linear or non-linear relationship of SQMs with LMR metrics. In other words, if our chosen regression tool estimates are not good then we would observe a high rate of false positives and error. For analyzing the effectiveness of our piece-wise linear model, we examine how closely Veracity can estimate the SQMs for the given *instantaneous* LMR metrics. We also show that piece-wise linear regression is comparable to linear ridge regression [8] when SQMs have a linear relationship with LMR metrics but outperforms it when they have a non-linear relationship.

We select 15 days' worth of data for conducting the analysis. We started with around 583 LTE cell towers and applied our median-based anomaly detection test (from section III-A) to filter out anomalous cell towers. This is done because operational networks have a large number of network events such as planned upgrades that can significantly impact the SQMs. Thus, after applying our filter, we identify 69 non-anomalous cell towers for *Accessibility*, 51 cell towers for *Retainability* and 34 cell towers for *Downlink Throughput*.

Now, we present error percentages for both ridge and piece-wise linear regression (note that we apply ARESlab for piece-wise linear regression). In ARESlab implementation, the number of breakpoints depends on the number of observation points, LMR metrics and a control parameter c. c captures the Generalized Cross-Validation (GCV) penalty per breakpoint. Larger values of c will lead to fewer breakpoints. For both ridge and ARESlab, we start by building a global model on the entire dataset, instead of building a model for each cell tower separately. In case of ARESlab, after the training phase, we use

Regression	Retainability	Accessibility	DL Throughput
Ridge	0.75%	0.13%	12.84%
Piece-wise Linear	0.79%	0.13%	7.84%

TABLE III
NORMALIZED ABSOLUTE MEAN ERROR.

SON Feature Trials	Veracity matches Ops conclusion (Litmus results used by Ops)	Number of quality metrics
A	8	8
B	5	8

TABLE IV
SUMMARY OF SON FEATURE TRIAL RESULTS.

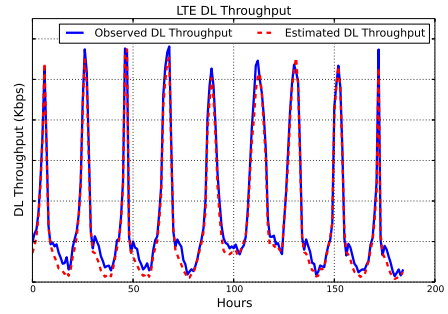
Veracity to estimate SQMs for different values of c (ranging from 1 to 14) per cell tower. Increasing c beyond 14 resulted in the number of breakpoints going down to one or zero (*i.e.*, model has only constant term). We then compute normalized absolute mean error per cell tower for different values of the control parameter c. Different cell towers have minimum error for different values of the control parameter. We took the minimum error for each cell tower. In case of ridge, we use the model to estimate SQMs and then compute normalized absolute mean error per cell tower. We tabulated results for both ridge and piece-wise linear (ARESlab) in Table III. It is evident from the table III that when a SQM shares a linear relationship with LMR metrics, piece-wise linear regression is comparable to ridge regression [8]. However, piece-wise linear outperforms ridge when the relationship becomes non-linear like in the case of DL throughput. Thus, piece-wise linear regression effectively captures the relationship between SQMs and LMRs.

Finally, we apply statistical change detection across all the SQMs and successfully confirm that the *estimated* SQMs closely resembles the *observed* SQMs. This yields no false positive and proves that our piece-wise linear regression is a good model for operational cellular network data.

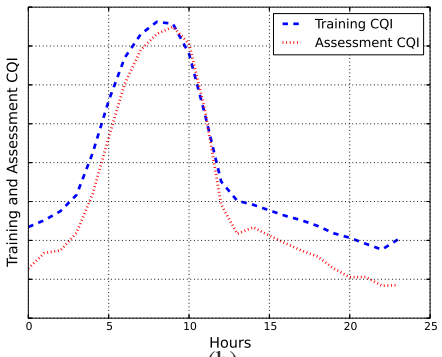
B. SON Feature Trials

We now evaluate Veracity during two SON feature trials in both LTE and UMTS operational cellular networks. Each of the feature trials spanned multiple days on a large number of cell towers (129 LTE eNodeBs and 349 UMTS NodeBs). During the trials, multiple SON functionalities were automatically executed resulting in an order of hundreds of thousands of configuration changes (or SON actions). We use Veracity for post-hoc analysis of the service performance impact of SON actions. The assessments for these two SON feature trials were already conducted in the past by the Engineering teams using Litmus [14]. In this section, by comparing Veracity to Litmus, we demonstrate the usefulness of Veracity and highlight case scenarios where Litmus would have been inaccurate in analyzing the impact.

We summarize our results in Table IV. We call a performance assessment to be a match to the operations conclusion if Veracity concludes the same impact as Litmus. For both feature trials, we have a total of 8 quality metrics to analyze (LTE accessibility, LTE retainability, LTE downlink throughput, UMTS voice accessibility, UMTS voice retainability, UMTS data accessibility, UMTS data retainability, UMTS downlink throughput). As observed from Table IV, for trial A, we have a good match with operation's conclusion using Veracity. However, for trial B, we have only 5 out of 8 cases that aligned with operation's conclusion. On further investigation and interactions with operations and engineering teams, we find that Veracity output is more accurate than Litmus. Due to space limitation, we will only present explanation for one of three mismatch cases.



(a)



(b)

Fig. 3. (a) shows relative improvement in LTE downlink throughput during feature trial B, correctly captured by Veracity. (b) shows degradation in channel quality indicator during the assessment interval as compared to training interval.

SON configuration changes induced a statistical improvement in LTE downlink throughout for trial B which is correctly captured by Veracity whereas Litmus found no statistical change. Figure 3(a) shows the *observed* LTE downlink throughput as measured in field and the *estimated* throughput using the piece-wise linear model in Veracity. We observe that the *observed* SQM values are higher than the *estimated*. The peaks in graph are during weekends (non-working hours) and the valleys are during the weekdays (working hours). After careful screening of the corresponding LMR metrics, we notice that the LTE radio environment indicator (CQI) degraded after the SON functionality was turned ON (see Figure 3(b)). For a degraded CQI, we expect downlink throughput to degrade, which is correctly reflected in our *estimated* throughput. We confirm this with the radio experts. Despite the fact that downlink throughput does not change in the absolute numbers, we can infer a relative improvement because of the SON actions. Veracity indeed captured this behavior that was previously missed by Litmus.

V. Operational Experiences

In this section, we demonstrate two case studies where the Network Engineering and Operations team at a US-based cellular service provider were testing Mobility Load Balancing SON function and used Veracity to assess the performance of selected professional football events. We show that Veracity can effectively quantify the impact of SON on SQMs.

Mobility Load Balancing (MLB) is often particularly important for large facilities that manifest large concentration of users with unusually large and irregular surges of traffic – such as in stadiums, convention centers, airports, concert halls, theme parks and festival attractions. Network and radio resources need to be well managed to achieve the most efficient traffic load distribution. Yet, predetermined cell selection priorities among different carrier frequencies and the inherent coverage discord due to power and mobility settings can easily lead to load imbalance. For instance, LTE carriers are assigned with higher cell selection priorities than UMTS frequencies, forcing all LTE enabled UEs (such as mobile phones) to camp on LTE instead of UMTS. In addition, lower carrier frequencies have stronger penetration than higher carrier frequencies, capturing more users onto say the 700 MHz band than onto the 1900 MHz band.

The tremendous traffic irregularity at such locations typically aggravates the problem. In the instance of sporting events, the traffic patterns are highly volatile – traffic load depends on the progress of the game, the total attendance, as well as factors including attendees enthusiasm, weather conditions (e.g., cold weather may discourage user from using their devices). The usage profile at such locations is also unusual with possibly more upload traffic than download because the users are more likely to share pictures and videos via social networks.

In such scenarios, the expectation is to balance the traffic load across all the available carrier frequencies within a technology (Intra Radio Access Technology (RAT)) as well as across technologies (Inter RAT), so as to achieve increased system capacity along with better end user service performance quality experience. The SON MLB function facilitates this desired outcome by manipulating the RAN mobility parameters and selectively instructing some UEs to switch from high loaded carrier to low loaded carrier. The condition to trigger offloading of traffic and the trigger thresholds is left to vendor implementation.

After the function activation if the estimated values are significantly greater than the observed values, then we conclude that the activation resulted in degradation. On the other hand, if the observed values are significantly greater than the estimated values, then we conclude improvement. Due to page limit, we will present a subset of graphs on the SQMs of interest. The two vertical lines on the graphs indicate the event assessment period. No significant difference indicates no impact.

A. SON assessment at stadium X

In our first case study, we demonstrate that our approach can accurately assess the performance impact of SON function activation. Using our approach, we compare the SQMs at the cell towers affected by the SON activation with the SQMs computed. Figure 4 shows the time-series for the LTE retainability observed at the fields (*observed retainability*) and LTE retainability provided by Veracity (*estimated retainability*). Using Veracity, we observe an improvement in LTE *Retainability* and LTE *Accessibility*, whereas LTE *DL throughput* and UMTS quality metrics show no significant difference – the SQMs being in alignment with estimation.

With respect to the LMR metrics in our data model, we find that the traffic volume metric has comparable values in training and assessment intervals (shown in Figure 4(b)), whereas the downlink block error rate (DL BLER) is distinctly higher in assessment interval than in training interval, indicating an increased interference (shown in Figure 4(c)). If SON activation didn't perform as required then we would have observed degradation in SQMs because of increased interference in the assessment interval. Hence, we conclude that the enabling of the SON function has benefited the *Retainability* metric as the service quality improved under a more adverse radio interference environment.

B. SON assessment at stadium Z

In our final case study, we compare the outcome of Veracity with existing single time-series detector system (Mercury [15]) used by the Networks and Operations team for quantifying the effectiveness of system feature change – the enabling of SON controller functions in this case. Figure 5 shows the comparison of Veracity and the single time-series detector Mercury for LTE *Accessibility*. If we use the existing single time-series detector only analysis, we would conclude that the performance degraded because of SON activation. However, close examination finds that because Mercury quantifies each SQM behavior based on just that metric's past baseline without considering the underlying LMR metrics, it concludes LTE *Accessibility* has degraded. However, the degradation in LTE *Accessibility* is expected due to an increase in the traffic load during the assessment interval and is correctly quantified by Veracity as having no impact. The drop in accessibility due to an increase in the traffic during assessment interval was falsely attributed to the system change by single time-series detector system like Mercury. Thus, Mercury had falsely attributed the degradation to the system feature change, whereas Veracity accurately captured the expected degradations and labeled as no impact.

VI. Related Work

Impact assessment of network changes: Our work relates to the area of change impact analysis of cellular networks. The key idea is to compare performance before and after a change and label the impact as either improvement, degradation or no change. Mercury [15] uses rank-based cumulative sums to identify statistical changes in performance after a major network upgrade. Prism [13] offers a near real-time assessment of performance and uses robust singular value decomposition to identify anomalies in performance. Spectroscope [19] and X-ray [4] compare two executions of program before and after the change to diagnose performance changes. FUNNEL [26] uses

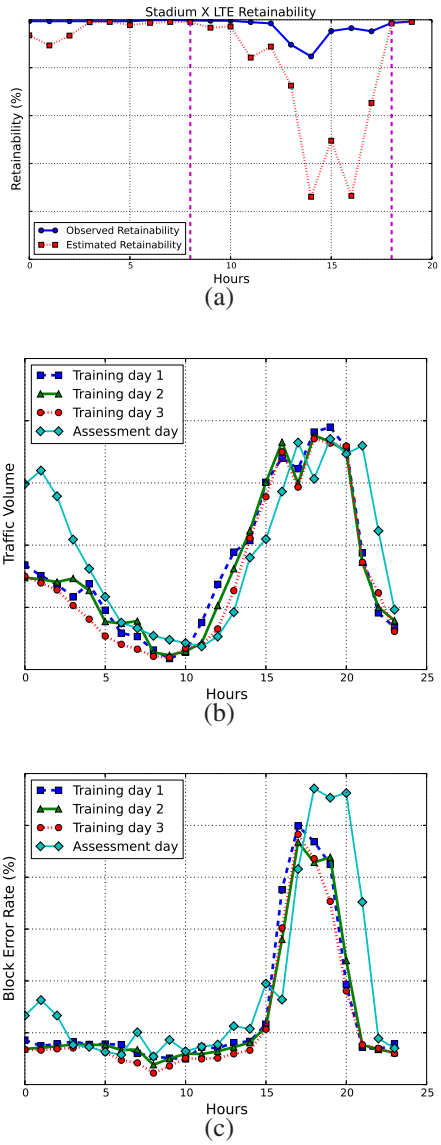


Fig. 4. (a) shows that the LTE data retainability improved during game at stadium X due to SON function. (b) shows that the LTE traffic is comparable in both assessment and training intervals. (c) shows that the block error rate is higher in assessment interval.

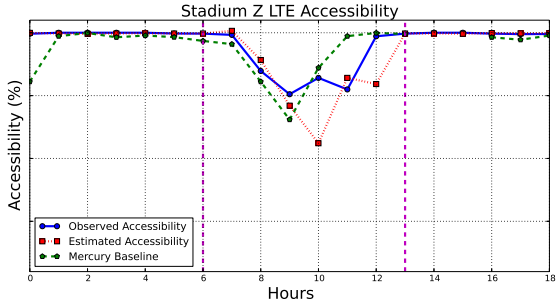


Fig. 5. At stadium Z, LTE accessibility has no impact and successfully captured by Veracity but flagged as degradation by traditional approaches.

Difference in Difference approach to detect the performance impact of software changes deployed in large Internet-based services. Litmus [14] uses performance comparisons between study group (network elements with change) and control group (network elements without the change) to extract out the effect of external factors and quantify the performance impact on the study group. All of these techniques construct the baseline using a before-interval time-series and lack a holistic view across multiple metrics. This is not suited well for quantifying the performance impact of SON.

Modeling cellular network measurements: In recent years, there has been a lot of research on analyzing and modeling performance of cellular networks [5], [7], [9], [16], [17], [20], [21], [23], [25], [27]. Xu *et al.* [25]reveals the fundamental differences between cellular data networks and the wireline networks. Shafiq *et al.* [20] illustrates how user population and behavior during crowded events and venue locations result in significant voice and data performance degradation. Shafiq *et al.* [21] presents the study of Internet traffic dynamics of cellular networks. [5] models the relationship between web quality of experience and factors such as signal strengths, load and handovers. [27] uses machine learning techniques such as Adaboost to learn the root-causes of call drops. [12] proposes to use the correlation among SON performance metrics to better diagnose problems in the radio access network. All the above papers analyze the traffic dynamics, measurements needed for performance analysis and users' quality-of-experience.

VII. Conclusions and Future Work

In this paper, we presented Veracity, a new model-driven approach to quantify the impact of SON (Self-Organizing Networks) actions on service performance in cellular networks. It effectively accounts for the dynamic nature of SON and accurately captures the impacts across multiple SQMs. We used Multivariate Adaptive Regression Splines (AERSlab) to capture the relationship of the underlying LMR metrics with SQMs. Our results demonstrate that the model-driven approach in Veracity does not yield false positives. Hence, it is more effective than previous time-series based approaches. Veracity is now being used successfully by the field operation teams of the cellular service provider for analyzing the effectiveness of SON deployed in event locations such as game stadiums and arenas. In the future, we plan to apply Veracity to determine whether SON is unnecessarily flip-flopping changes multiple times to reach the optimal solution or if SON is failing on executing required configuration changes. As operational networks start transitioning into software-defined networks, it would be interesting to incorporate Veracity in the feedback loop of the SON controllers.

Acknowledgment

We thank Kurt Huber, Ioannis Broustis, Jennifer Yates, our shepherd Abdelouahed Gherbi and the CNSM anonymous reviewers for their insightful feedback on the paper. We strongly appreciate the collaboration and continuous support from the Network Engineering and Operations teams in the application of Veracity, regular feedback to improve its usability and case-study analysis.

REFERENCES

[1] 3GPP LTE TS 32.450. Telecommunication management; Key Performance Indicators (KPI) for Evolved Universal Terrestrial Radio Access Network (E-UTRAN): Definitions.

[2] 3GPP LTE TS 32.500. Telecommunication management; Self-Organizing Networks (SON); Concepts and requirements.

[3] 3GPP LTE TS 36.300. Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Overall description.

[4] M. Attariyan, M. Chow, and J. Flinn. X-ray: Automating root-cause diagnosis of performance anomalies in production software. In *USENIX OSDI*, 2012.

[5] A. Balachandran, V. Aggarwal, E. Halepovic, J. Pang, S. Seshan, S. Venkataraman, and H. Yan. Modeling web quality-of-experience on cellular networks. In *ACM MOBICOM*, 2014.

[6] N. Feltovich. Nonparametric tests of differences in medians: Comparison of the wilcoxonmann-whitney and robust rank-order tests. *Experimental Economics*, 2003.

[7] A. Gember, A. Akella, J. Pang, A. Varshavsky, and R. Caceres. Obtaining in-context measurements of cellular network performance. In *ACM IMC*, 2012.

[8] A. E. Hoerl and R. W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 1970.

[9] J. Huang, Q. Xu, B. Tiwana, Z. M. Mao, M. Zhang, and P. Bahl. Anatomizing application performance differences on smartphones. In *ACM MobiSys*, 2010.

[10] G. Jekabsons. ARESLab: Adaptive Regression Splines toolbox for Matlab/Octave, 2011. available at http://www.cs.rtu.lv/jekabsons/.

[11] J. R. Lanzante. Resistant, robust and non-parametric techniques for the analysis of climate data: Theory and examples, including applications to historical radiosonde station. *International Journal of Climatology*, 1996.

[12] P. M. Luengo, I. de la Bandera Cascales, E. Khatib, A. G. Andrades, I. Serrano, and R. Barco. Root cause analysis based on temporal analysis of metrics toward self-organizing 5g networks. *IEEE Transactions on Vehicular Technology*, 2016.

[13] A. Mahimkar, Z. Ge, J. Wang, J. Yates, Y. Zhang, J. Emmons, B. Huntley, and M. Stockert. Rapid detection of maintenance induced changes in service performance. In *ACM CoNEXT*, 2011.

[14] A. Mahimkar, Z. Ge, J. Yates, C. Hristov, V. Cordaro, S. Smith, J. Xu, and M. Stockert. Robust assessment of changes in cellular networks. In *ACM CoNEXT*, 2013.

[15] A. Mahimkar, H. H. Song, Z. Ge, A. Shaikh, J. Wang, J. Yates, Y. Zhang, and J. Emmons. Detecting the performance impact of upgrades in large operational networks. In *ACM SIGCOMM*, 2010.

[16] A. Nikravesh, D. R. Choffnes, E. Katz-Bassett, Z. M. Mao, and M. Welsh. Mobile network performance from user devices: A longitudinal, multidimensional analysis. In *PAM*, 2014.

[17] A. Nikravesh, H. Yao, S. Xu, D. Choffnes, and Z. M. Mao. Mobilyzer: An open platform for controllable mobile network measurements. In *ACM MobiSys*, 2015.

[18] J. Ramiro and K. Hamied. Self-organizing networks (SON): Self-planning, Self-optimization and Self-healing for GSM, UMTS and LTE.

[19] R. R. Sambasivan, A. X. Zheng, M. D. Rosa, E. Krevat, S. Whitman, M. Stroucken, W. Wang, L. Xu, and G. R. Ganger. Diagnosing performance changes by comparing request flows. In *USENIX NSDI*, 2011.

[20] M. Shafiq, L. Ji, A. Liu, J. Pang, S. Venkataraman, and J. Wang. A first look at cellular network performance during crowded events. In *SIGMETRICS*, 2013.

[21] M. Shafiq, L. Ji, A. Liu, and J. Wang. Characterizing and modeling internet traffic dynamics of cellular devices. In *SIGMETRICS*, 2011.

[22] S. Siegel and N. J. J. Castellan. *Nonparametric Statistics for the Behavioral Sciences*. New York: McGraw-Hill, 1998.

[23] G.-H. Tu, Y. Li, C. Peng, C.-Y. Li, H. Wang, and S. Lu. Control-plane protocol interactions in cellular networks. In *ACM SIGCOMM*, 2014.

[24] Wikipedia. Polynomial regression. https://en.wikipedia.org/wiki/Polynomial_regression.

[25] Q. Xu, J. Huang, Z. Wang, F. Qian, A. Gerber, and Z. Mao. Cellular data network infrastructure characterization and implication on mobile content placement. In *SIGMETRICS*, 2011.

[26] S. Zhang, Y. Liu, D. Pei, Y. Chen, X. Qu, S. Tao, and Z. Zang. Rapid and robust impact assessment of software changes in large internet-based services. In *ACM CoNEXT*, 2015.

[27] S. Zhou, J. Yang, D. Xu, G. Li, Y. Jin, Z. Ge, M. B. Kosseifi, R. Doverspike, Y. Chen, and L. Ying. Proactive call drop avoidance in umts networks. In *Proc. of IEEE INFOCOM*, 2013.