Big data-empowered system for automatic trouble ticket generation in IoT networks

David Palacios, César Morillas, Manuel García and Pablo Tapia
Tupl Spain SL
Tupl Inc.,
Málaga, 29010, Spain.
{david.palacios, cesar.morillas, manuel.garcés}@tupl.com, pablo@tupl.io

Abstract—One of the major concerns for a mobile network operator is the customer trouble ticket (TT) management, as it is directly related to the customers’ satisfaction about the provided quality of service. This issue becomes specially relevant when the number of end devices being served by the cellular network increases dramatically. Such is the case of communications under the Internet of Things (IoT) paradigm. In this case, a traditional approach for TT management appears as insufficient and novel mechanisms for TT management need to be devised. This work proposes a big data-empowered framework for automatic TT generation, showing its capability to deal with such amount of data. Results in a live network providing IoT communication show that the proposed framework was able to identify the network anomalies that were identified following a traditional approach as well as many others that remained hidden under that approach.

Index Terms—Internet of Things (IoT), root cause analysis (RCA), anomaly detection, trouble ticket (TT), big data.

I. INTRODUCTION

The automatic management of cellular networks has greatly evolved over the last years with the concept of self-organizing networks (SON) [1]. However, some tasks for network management, specially under the scope of network self healing [2], still mostly depend on human action. This is the case of customer trouble tickets (TTs). As of today, different examples can be found on the usage of TT to delve into the network state for anomaly detection and root cause analysis of network failures [3]–[5]. However, fewer efforts have been put into the automatic generation and management of customer TTs. In this regard, works like [6] and [7] could be highlighted. In [6], a method to generate and enrich TTs using network performance data is proposed. In [7], a method to correlate and hierarchically aggregate TTs according to their similarity is proposed.

Together with this, with the advent of the fifth-generation (5G) of mobile networks, new and different service categories will be covered, extending both the casuistry of network faults and the traffic profiles, and thus, making even more difficult automating tasks, such as TT generation and management.

Within the communications types to be covered in 5G, massive machine-type communications (mMTC) hold a special place. Regarding this, two main differentiating aspects may be highlighted with respect to traditional communications. First, they aim at simultaneously providing service to a huge amount of devices, leading to very high data volumes for both the user and the control planes. And second, the offered traffic profile for mMTC greatly differs from that of traditional communications. Whereas this latter is characterized by a smooth and seasonal profile, the former usually consists in a bursty pattern [8], which hinders its automatic management, leading to a high number of false positives in automatic anomaly detection tools.

As a result, and pushed by the advances in computation power and processing architectures, new approaches for automatic management of cellular networks are gaining momentum. This is the case of big data, appearing as an enabler for next-generation SONs ( [9], [10]) and, within these, self-healing functions [11].

Thus, the main contributions of this work are:

- The proposal of a big data-empowered framework to automatically generate TTs for traditional communications and mMTC.
- The usage of anomaly detection and RCA tools to automatically indentify and classify the network state, eventually used to enrich TTs.
- The ability to aggregate low-level TTs into global or master TTs, according to their similarity.
- The usage of call detail records (CDRs) as the data source for network and service performance allowing both a batch and a streaming way of operation. CDRs are a very detailed and low-level source of information, which, through aggregation, allows performing a RCA at different hierarchical levels.

This way, the proposed framework appears as a fully unattended and end-to-end scheme for TT generation, outpacing [6], in which network performance data must be explicitly provided, and [7], in which only a small part of the pipeline is covered: the TT aggregation.

II. FRAMEWORK FOR AUTOMATIC TT GENERATION

The proposed framework follows the flow diagram shown in Figure 1. First, in a data ingestion stage, customer-centered performance data are gathered from the cellular network and subsequently aggregated in enriched data blocks, according to different criteria. Next, key performance indicators (KPIs) are generated from these latter by applying both a statistical analysis and an event processing logic on them. After KPIs
have been generated, two parallel steps take place: a feature service block and an anomaly detection block, both feeding a trigger logic block. The objective of these three blocks is double. On the one hand, they aim at shaping the performance data into a format (features) that optimizes the operation of a subsequent block for root cause analysis (RCA). On the other hand, they aim at identifying which of the samples correspond to actual performance degradations or anomalies. The RCA block would then be in charge of determining the cause behind such anomaly by assessing the features at its input. Finally, each anomaly would end up in a TT, enriched by its underlying cause. These would reach a ticket service block, which would eventually managed the so created tickets.

These functional blocks are further described in the following sections.

Fig. 1. Flow diagram for the proposed framework for automatic TT generation.

A. Data ingestion

The data resulting from the data ingestion stage are stored in a big data structured database with high reading and writing capacities, given the large volume of radio and network core elements being monitored and the streaming nature of the data being stored. Besides this, the deployed subsystem for data ingestion is able to perform online transformations on raw data, showing a scalable architecture by parallelizing the process of independent network elements.

The data ingestion tool is finally able to group samples into batches, so that this approach can be followed for subsequent stages.

B. KPI calculation from input data

This stage aims at providing a set of performance metrics from raw data by applying a two-step transformation:

- CEP metrics [12]: The objective of this block is to produce counters after parsing both numerical and non-numerical information from the CDRs. To that end, counters derived from a complex event processing (CEP) block over the incoming flow of CDRs are defined. Some examples of these counters could be:
  - "Total number of DIAMETER/RADIUS CDRs".
  - "Number of failed DIAMETER/RADIUS CDRs".
  - "Number of failed RADIUS CDRs due to lack of IP addresses".

All these counters are subsequently aggregated in real time at different levels. Some possibilities are:
  - Network core elements.
  - Network radio elements.
  - Subscribers and corporate accounts.
  - Terminal types (e.g., phone vendor and model).
  - Countries.

- KPI generation: taking the previously defined aggregated counters as inputs, a second process block is applied. This block is in charge of computing a number of performance metrics, such as rates defined over counters. Some examples of the so defined KPIs are the following:
  - "DIAMETER CDR failure rate", defined as the "Number of failed DIAMETER CDRs" over the "Total number of DIAMETER CDRs".
  - "RADIUS failure rate due to the lack of IP addresses", defined as the "Number of failed RADIUS CDRs due to lack of IP addresses" over the "Total number of RADIUS CDRs".

C. Feature service

The purpose of both this functional block and the anomaly detection one is to create discrete-value variables, called features onwards, from continuous information (i.e., KPIs). This discretization aims at optimizing the performance of subsequent stages and allowing the confirmation of potential network performance degradations as eventual anomalies.

In particular, the discretization carried out in this functional block pursues integrating the troubleshooting expert’s knowledge into a number of categories, by wisely choosing the conditions that allow differentiating KPIs into a number of categories, such as low, medium or high value.
This functional block allows defining different types of features; namely:
- Range features. They are built by segmenting the whole range of values of a certain KPI into few categories. For example, a ratio KPI could be split into low (from 0% to 75% of its range), medium (from 75% to 95%) and high (from 95% to 100%).
- Time features. The resulting features depend on the historical evolution of one or several KPIs. For example, a feature could be considered as high if a certain KPI achieved a given steady value during a time window.
- Complex features. These are features which could neither be implemented following a range, nor a time approach. In this case, features involving a number of KPIs under different and complex conditions would be generated. The anomaly detection block could be considered as a particular case of complex feature generator, which is described in the next section.

D. Anomaly detection and trigger logic

The anomaly detection system operates on a sample basis, following a streaming approach, and evaluating the difference between the current sample \(x[n]\) and its expected value \(\hat{x}[n]\). To do this, different thresholds are defined. These thresholds allow identifying whether a new sample under assessment is within the range of expected values, thus, identifying it as normal; or, on the other hand, is out of range, identifying the sample as an anomaly.

In particular, different dimensionless threshold levels are defined, in order to classify the anomaly into different severity levels. This approach provides higher flexibility, allowing the framework to be used in a general use case, rather than limiting it to a specific environment.

Cellular networks are systems in which behaviors with different time resolutions can be simultaneously found. On the one hand, metrics coming from the time aggregation of several users or network nodes often lead to rapidly varying signals, whose instantaneous evaluation could be misleading. To cope with this, the expected value for a given sample is computed by means of a finite impulse response (FIR) low pass filter, implemented as a moving average over past samples, using a sliding time window. On the other hand, and since cellular networks have a strong periodic behavior, many of the metrics quantifying their performance inherit this periodicity. To take this into account and minimize its effect on many of the metrics quantifying their performance, a sliding time window is split and spanned over a number of periods. Figure 2 jointly shows the sample to be analyzed, the filter which delimits the samples of interest and the resulting subset of filtered samples, \(\hat{x}[n]\):

![Image](image_url)

Fig. 2. Definition of \(\hat{x}[n]\) as a subset of \(x[n]\), after filtering this latter.

The expected value for the variable under assessment, \(\hat{x}[n]\), is computed as the mean value for \(\hat{x}\), \(\mu_{\hat{x}}\). Finally, to compute a dimensionless value for the difference of \(x[n]\) and \(\hat{x}[n]\), this difference is normalized by the standard deviation of \(\hat{x}\) (\(\sigma_{\hat{x}}\)). This results in \(d[n]\) (eq. (2)), which is the magnitude to be finally compared against the thresholds:

\[
d[n] = \frac{x[n] - \mu_{\hat{x}}}{\sigma_{\hat{x}}} \quad (2)
\]

The result of this is a feature with a discrete value, which indicates the severity of the observation for a given KPI, depending on the threshold settings.

The sample under assessment will be eventually considered as an anomaly if a set of conditions (i.e., trigger logic) are met considering all the features defined. If several anomalies take place within a given time window for a certain aggregation level (Section II-B), they could be considered as a single anomaly, which will be referred to as detection onwards.

E. Root cause analysis

The purpose of the functional block for root cause analysis is to determine the underlying cause of an abnormal behavior, so that the TT to be generated can be enriched with this information.

After a sample is determined as abnormal, the root cause for this behavior is sought. This is done by means of a functional block for root cause analysis, which, in this work, is implemented following a semi-supervised approach (e.g., [13]). That is, first, and in order to build the training set for a subsequent supervised technique for classification, a clustering method is used to find underlying patterns among the samples considered as anomalous. Specifically, a k-means clustering algorithm is used [14], given its implementation simplicity and robustness. Next, once that a number of anomalous samples...
have been grouped into clusters showing similar behavior patterns, a troubleshooting expert labels the so defined clusters, according to their statistical behavior. Finally, a random forest classifier is used to implement the functional block for root cause analysis, given its ability to overcome overfitting and dealing with both numerical and categorical features [15].

F. Ticket service

At this stage, every abnormal sample has a label attached to it, stating the reason to consider such sample as abnormal, as well as a recommended action. This label will help with the TT management, as it is shown in Figure 3. As it is shown in this flow diagram, whenever a labeled incidence arrives from the classifying system, a TT is created, containing information about its root cause (TT_{N}.RC, for the Nth TT), about the anomaly start time (TT_{N}.startTime) and about the anomaly duration (TT_{N}.duration).

At this point, a high amount of TTs may have arisen as a result of the detection and classification stages. However, it is likely that many of such tickets (and thus, anomalies) correspond to a single event in the network, which triggered different performance degradations. Thus, in order to reduce the amount of TTs that may arise from the procedure so far, a stage for TT aggregation has been devised. In this case, the aggregation is done according to the time dimension, as well as the topology and root cause behind each TT.

After single TTs have been merged into master TTs, an additional check is performed. If the anomaly behind this master TT ceases during a certain time above a user-defined time threshold, the corresponding master TT is automatically closed, not requiring further actions. If, on the other hand, this anomaly persists, the master TT is forwarded to the network operation center (NOC), together with the recommended action to recover the normal behavior of the network.

III. PROOF OF CONCEPT

A. Experiment setup

The proposed framework was assessed in a live environment: a mobile network, providing Internet of Things (IoT) access to 8 countries, more than 750 Access Point Names (APNs) and more than 1200 corporate accounts that involve more than 7.5 million devices during seven weeks, where its performance was compared with the result of a classical operation. In this environment, different network issues were considered: all of them related to three network subsystems: the Remote Authentication Dial-In User Service (RADIUS), the DIAMETER service for authentication, authorization and accounting (AAA) and the short messaging service (SMS).

Two kinds of fault causes were roughly considered:

- Those which resulted in an anomaly in the time evolution of a set of KPIs. For example, only a 10% of the expected number of connections are registered for a given user.
- Those which resulted in KPIs showing unhealthy values, according to the troubleshooting expert’s knowledge. For example, the number of failed connections in a given network element is higher than 1%.

The data ingestion stage was orchestrated with Apache NiFi, Kafka and HBase databases. The so created pipeline was in charge steadily monitoring the network performance data with a time resolution of 4 minutes. Some examples of the KPIs which resulted from this CDRs are the number of SMS-related CDRs or the rate of RADIUS connections rejected due to the lack of IP addresses.

Following the KPI generation, both the feature service and the anomaly detection stages were executed. Regarding the former, features like the discretized rate of rejected RADIUS connections were computed, considering up to a 0.5% as a low value; up to a 1% as a medium value, and above 1% as a high value, in this case. Regarding the anomaly detection stage, T was set to 5 days; M was set to 3 samples, and P was set to 360 samples, corresponding to 24 hours. Two different thresholds...
were set: 3 and 5, meaning that the difference between $x[n]$ and $\mu_2$ was compared with $3\sigma_2$ and $5\sigma_2$, respectively. This allowed differentiating among three degradation levels: low, medium and high. Next, a trigger logic was applied over the features defined in the previous steps to finally identify network anomalies.

In the following step, the random forest classifier was applied to determine the root cause of a given anomaly for the subsequent TTs to be enriched. In this case, the criterion used for the information gain calculation was gini; the maximum depth of a three was set to 5 and the number of trained and ensembled trees was 20.

Regarding the stage for TT management, the time window within which a previously detected anomaly must cease to consider this as resolved (and thus, to automatically close the corresponding master TT) was 12 minutes.

B. Results and discussion

In the course of the seven weeks of network monitoring, 1851 detections were registered, after the trigger logic was applied. That is, 1851 single TTs, enriched with the class labels which the classifier determined. As a result of the TT management stage, 1060 master TTs were eventually produced, covering all the master TTs that were generated following the classical approach and successfully identifying real problems in the cellular network. This demonstrated the ability of the proposed framework to automate this task.

Figure 4 shows the relative frequency of the identified root causes. As it can be seen, most of the identified anomalies were related to an abnormal number of CDRs, both for the RADIUS and for the DIAMETER services.

On the other hand, Figures 5 and 6 show two cases in which an undetected anomaly was identified by the proposed framework, but remained unnoticed following the traditional approach for network performance inspection. Thus, these two cases resulted in two master TTs, which were afterwards delivered to the corresponding department in the operator’s NOC. Figure 5 shows an example of an anomaly triggered by the feature service. In this case, the rate of terminated DIAMETER sessions due to administrative issues reached a first threshold of 0.5% (leading to a medium severity anomaly) to eventually exceed the threshold delimiting a severe anomaly (1%). Figure 6, on the other hand, shows an example of an anomaly due to an unexpected trend in the per user dowlink data volume, which triggered the anomaly detection block. One of the reasons why these behaviors remained unnoticed following the traditional approach for the inspection of network performance and TT generation is the amount of performance data being steadily generated in a cellular network, which makes its inspection become an overwhelming task. This issue is particularly relevant and concerning in the field of IoT communications, as shown in this work, in which the amount of end devices is specially high.

IV. Conclusions

In this work, a framework to automatically generate TTs has been proposed. To that end, a variety of network issues have been detected and used to enrich and group TTs, which would allow reducing their time response. All these issues were captured using a combination of an anomaly detection algorithm and a set of rules based on expert knowledge.

The list of automatically identified anomalies includes all the ones detected by the classical approach, proving that the proposed framework is able to cope with the operator’s objective of automating the TT generation from the very moment that a relevant issue happens to any customer following a streaming approach.

REFERENCES

[2] 3GPP, “Self-Organizing Networks (SON); Self-healing concepts and requirements, version 14.0.0 (2017-04),” TS 32.541.
Fig. 5. Example of detected anomaly: KPI above allowed values for healthy network operation.

Fig. 6. Example of detected anomaly: abnormal trend for downlink data volume.


