Root Cause Analysis Based on Temporal Analysis of Metrics Toward Self-Organizing 5G Networks

Pablo Muñoz, Isabel de la Bandera, Emil J. Khatib, Ana Gómez-Andrades, Inmaculada Serrano, and Raquel Barco

Abstract—By 2020, mobile networks will support a wide range of communication services while at the same time, the number of user terminals will be enormous. To cope with such increased complexity in network management, innovative solutions for the next generation of self-organizing networks (SONs) need to be deployed. In the field of self-healing, the heterogeneous character of future fifth-generation (5G) radio access networks (RANs) will provide a diversity of measurements and metrics that can be translated into valuable knowledge to support detection and diagnosis activities. The more complete the information gathered, the better the SON mechanisms will be able to effectively analyze and solve radio problems. However, temporal dependence between metrics has not been previously addressed in the literature. In this paper, a self-healing method based on network data analysis is proposed to diagnose problems in future RANs. The proposed system analyzes the temporal evolution of a plurality of metrics and searches for potential interdependence under the presence of faults. Performance is evaluated with real data from a mature Long-Term Evolution (LTE) network. Results show that the proposed method exploits the available data in the context of heterogeneous scenarios, reducing the diagnosis error rate.

Index Terms—Correlation, fault diagnosis, Long-Term Evolution (LTE), self-healing, self-organizing networks (SONs).

I. INTRODUCTION

The increasing complexity of the architecture of cellular networks toward the fifth generation (5G) [1] has complicated network management. Self-organizing networks (SONs) are those networks whose management is carried out with a high level of automation [2]. In the field of self-healing or automatic troubleshooting [3], the enormous diversity of performance indicators, counters, configuration parameters, and alarms has led operators to search for intelligent and automatic techniques that cope with faults in a more efficient manner, making the network more reliable [4]. A reference model for self-healing to unify terms, functions, and previous research is proposed in [5]. In general, self-healing has been focused on detection [6]–[9] and compensation [10]–[14] of cell outages in the radio access network (RAN), whereas less attention has been paid to more general faults [15]–[21]. In addition, some works addressing the self-healing in the core network [22]–[24] can be also found.

Diagnosis, which is also called root cause analysis, is a key function in fault management that allows the identification of the fault causes. Although some efforts have been devoted to the development of usable automatic diagnosis systems, existing solutions are commonly primitive approaches to diagnosis, whereas the more complex alternatives require a lot of information about the faults that is not available in most cases, e.g., the conditional probability density function of the metrics (symptoms) given the fault causes. Looking at the recent literature, in [25] and [26], an integrated detection and diagnosis framework to identify anomalies and find the most probable root cause is proposed. However, this approach is based on how much a metric should deviate from its usual behavior to be considered as degraded. The main drawback of this kind of approaches is that, in many cases, the effect on the metrics is not a clear deviation from the normal range but a small change (e.g., a peak or a step) in the temporal evolution of the metric that would be unfortunately disregarded since the method does not consider the temporal dependence of the metrics. In [27], the proposed method aims at determining the root cause based on clustering past fault cases, identifying recurrent network behavioral patterns. The adopted solution is problematic due to the use of thresholds. This commonly leads to a more drastic decision when determining whether a metric is degraded or not. In [28], a method for identifying the causes of changes in performance indicators by analyzing the correlation with a plurality of counters is also proposed. The diagnosis is carried out by means of classification/regression trees, which are used to predict membership of event counters in one or more classes of performance metrics of interest. However, this kind of solutions is typically based on fixed thresholds; hence, similar drawbacks as before are derived from this approach.

The diagnosis in cellular networks has been also approached by applying different mathematical techniques. In [18] and [29], Bayesian networks are proposed as the reasoning method for automatic diagnosis. However, the models must contain all the possible states of the network and their associated probabilities. In [30], self-organizing maps are utilized to help human experts in visualizing and grouping similarly behaving cells. This unsupervised technique is a reasonable starting point for the diagnosis in recently deployed networks, where a large number of identified faults associated with their symptoms are hard to get. As networks evolve, expert knowledge and
experience become key aspects to the development of automatic and more effective diagnosis systems than those based exclusively on self-organizing maps. For example, fuzzy logic combined with genetic algorithms has been applied in [31], where a rule base for troubleshooting is automatically built when a large database of previously diagnosed fault cases is available.

In this paper, an automatic diagnosis algorithm based on temporal analysis of the metrics is proposed. In particular, the similarity in time between metrics is evaluated and compared with stored patterns to determine the root cause of the fault. Metrics can be performance indicators, counters, mobile traces, alarms, configuration parameters, and even geographical data. The main contributions of the proposed algorithm are as follows.

- **The use of time series of metrics in self-healing.** In previous works, the input of the methods is normally given by a single value of each measured metric. This can be either an instantaneous value or an average value over a time period. As a consequence, valuable information (e.g., the time trend) may not be considered or even partially lost. In this paper, temporal dependence between metrics is taken into account through the analysis of time series of metrics instead of scalar values. This approach indicates an important change of philosophy in traditional techniques of fault diagnosis, particularly those based on exclusively measuring deviations from normal behavior.

- **The use of metrics collected from neighboring cells of the problematic cell.** In current solutions for automatic diagnosis, a cell is typically diagnosed from measurements and alarms gathered only in that cell. The diagnosis can be significantly improved by automatically giving more or less importance to metrics in a neighboring cell as a function of the overall impact of the fault on the neighboring cell. Moreover, in the context of heterogeneous deployments, the proposed method takes advantage of cells with large overlapping coverage areas (e.g., co-sited cells), where the metric correlations between overlapped cells are typically greater than in cells partially overlapped. Thus, if a cell is faulty, the impact on the neighboring cells is expected to be greater.

The proposed algorithm has been included in a patent application that has been filed on April 24, 2015 (application number PCT/EP2015/058924). The remainder of this paper is organized as follows. Section II formulates the problem and defines the system model. In Section III, the proposed algorithm is presented, describing in detail each of its functional blocks. Then, in Section IV, the algorithm is evaluated by using a collection of faults from a live network. Finally, Section V presents the main conclusions of this paper.

### II. System Model

Next-generation mobile networks will include a plurality of promising key wireless technologies. As a consequence, the network architecture will be heterogeneous. In Fig. 1, an example of these networks is represented.

As observed, the RAN is represented by a number of overlapping radio access technologies (RATs), such as High-Speed Packet Access or Long-Term Evolution (LTE). Within the same RAT, cells of different sizes (e.g., small cells) and frequencies can be also massively deployed. The use of base stations (BSs) with very different transmit power values and coverage areas and the coexistence of frequency bands with very diverse propagation conditions will surely involve a heterogeneous and complex environment from the perspective of network management. This network densification, together with device-to-device (D2D) connectivity, millimeter-wave transmission, and massive multiple-input–multiple-output (MIMO), will be also key features of the 5G technology to deal with the huge demand of traffic through an efficient utilization of available resources.

The future core network will be characterized by a centralized cloud computing-based architecture supporting current and future wireless communication standards. The core functional areas will be decomposed in virtualized network functions where control and user plane are implemented separately, allowing flexible deployment. In addition, the core network’s functions can be deployed either in a distributed manner at RAN nodes or centralized. For example, some core functionalities such as mobility management can be moved to the access nodes to reduce latency.

Since the previously described architecture is very complex and costly to manage, the operations, administration, and maintenance (OAM) system should enable intelligent and autonomous configuration, optimization, and troubleshooting for specific services and network nodes. In this sense, current SON solutions are still struggling to provide a comprehensive global network management [32]. To provide efficient management of the complete 5G system, the OAM has to collect information from all over the network. By exploiting the gathered information, network management functions are significantly improved. For example, in the context of fault management, a problem can be autonomously detected or diagnosed by using previously recorded data instead of traditional approaches based on simple thresholds. More specifically, in this paper, the proposed SON method is devoted to fault diagnosis in the RAN...
through data and correlation analysis techniques. This method can be located within the OAM system, and it will be connected with the fault detection, compensation, and recovery entities to receive information about the occurrence of a faulty situation and then compensate and fix the diagnosed problem.

The sources of information within the RAN are typically the BSs and the user equipments (UEs). As previously stated, the BSs control cells from different RATs, operating at different frequencies, having different coverage areas (e.g., macro- or small cells), etc. Statistics measured in the elements of the core network can be also used; however, such statistics should be aggregated at the cell level, which would require adding extra intelligence to the whole system. The information provided by the aforementioned sources can be classified into different categories.

- **Configuration parameters**: These represent the actual configuration of network elements and resources, for example, the maximum transmit power and the antenna downtilt of the BSs and the cell individual offset.
- **Counters**: This type of metric includes measurements from the network elements, which can be reported periodically or on-demand. Examples of counters are the number of seconds of cell availability, the number of successful handovers, and the number of dropped calls.
- **Key performance indicators (KPIs)**: These are derived from other measurements, e.g., through a specific formula of counters or other KPIs, with the aim of providing a meaningful performance measure. Examples of KPIs are the percentage of cell availability, the handover success ratio, and the call dropping ratio.
- **Alarms**: An alarm is a message generated by a network element when there is a failure. It may be triggered when some counter or KPI passes some threshold. For example, an excessive usage of a specific processor may trigger an alarm to indicate congestion issues.
- **Mobile traces**: This information involves measurements reported by the UEs that can be collected by the serving BS. The main advantage of the mobile traces is that the information is geolocated, providing deeper insights about radio problems in specific areas.
- **Drive tests**: This refers to field measurements, e.g., related to coverage and interference, performed in a certain area by specialized equipment, which can also provide precise location information. As human effort is required, drive tests are gradually being substituted by the use of mobile traces (i.e., minimization of drive tests).

On the one hand, this information can be used as input of the different phases of the troubleshooting process. In the detection phase, KPIs are useful metrics to determine the existence of problems because they measure the overall performance of some aspects of the network. Then, in the diagnosis phase, the other sources of information that provide more specific information can be used to determine the fault cause. In this paper, the proposed algorithm must be preceded by a fault detection phase, where a certain metric is detected as a potential symptom of an unknown fault.

On the other hand, in current networks, not all the information for fault management is collected and stored in databases due to technological constraints. Moreover, the available information is used only for manual and eventual queries by experts so that, once the problem is solved, the information is probably erased. With the advances in big data techniques and new technologies, the future 5G mobile networks are expected to be capable of processing and storing such huge amounts of information to provide innovative solutions for fault management.

### III. Method for Fault Diagnosis

The proposed method is an automatic diagnosis algorithm based on temporal analysis of network metrics. More specifically, the similarity between the time evolutions of these metrics is evaluated and compared with stored faults (i.e., a pair of symptoms and diagnosis) to determine the root cause of the fault. One of the inputs of the method should be a representative metric whose anomalous behavior has been found in a previous fault detection stage whose implementation could be based, for example, on a deviation of the metric or an abnormal trend over time. In any case, this metric is detected as a potential symptom of an unknown fault and called the primary metric hereafter. In cellular networks, KPIs related to accessibility (e.g., establishment success rate) or retainability (e.g., dropped calls rate) are typically used in the detection phase to monitor the health of the network. These kinds of metrics can be used as primary metrics since any degradation of these metrics could evidence the existence of a fault in the network.

Fig. 2 shows a high-level block diagram of the proposed method, where the main entities and variables are shown. These entities should be seen as black boxes whose implementation is not necessarily unique. Note that, for evaluation purposes, some aspects of these entities will be defined and implemented in this paper according to simple but effective models. The first step of the method (i.e., metric correlation) is the calculation of a correlation indicator that measures the similarity between the primary metric \(m_p\) and other metrics \(m_j\) from either the same cell or another. The reason for this is that, due to the occurrence of a fault at a certain time, some metrics can be simultaneously affected. As a consequence, these metrics will be potentially correlated in the time domain, and this dependence will enable the identification of the given fault. The selection of
metrics and neighboring cells is explained later. To compute the correlation values, a certain time interval of the metrics is considered, meaning that not only the current value of the metric is used but a vector of samples representing the temporal evolution as well. Thus, this approach overcomes the main drawbacks of the existing solutions. The result of this process is a vector $\mathbf{v}^{(p)}$ containing the respective correlation values. The next step (fault correlation) is the calculation of a weighted correlation indicator $v^{(p)}$ between the obtained vector $\mathbf{v}^{(p)}$ and past fault cases $\mathbf{S}^{(p)}$ stored in a database. The similarity between the current case and previous fault cases is determined to find the most likely cause of the problem. The weights $\mathbf{W}^{(p)}$ used for the calculation allow giving more or less importance to some metrics, depending on several aspects, such as the expert knowledge $\mathbf{E}$ or the overall impact of the fault on the respective cell. This vector of past fault occurrences will be updated by the learning entity. If the correlation value is not high enough, the cause of the problem will be unknown. The following sections explain the mathematical model of the system inputs and each of the functional entities previously presented.

A. Inputs of the Method

The input signals of the algorithm are the primary metric derived from the previous detection phase $\mathbf{m}_p$ and other metrics $\mathbf{m}_j$, which are commonly collected by the OAM and stored in a database. The metrics are collected at specific time intervals $L$ and stored in vectors of the same length $T$. For example, the configuration $L = 1 \text{ h } T = 24$ means that the samples of a certain counter are collected over each hour during 24 h, meaning that the value at 1 h is the number of events (e.g., dropped calls) measured in that interval. Then, vectors $\mathbf{m}_p$ and $\mathbf{m}_j$ can be formally defined as

$$\mathbf{m}_p = (m_{p,1}, m_{p,2}, \ldots, m_{p,T})$$

$$\mathbf{m}_j = (m_{j,1}, m_{j,2}, \ldots, m_{j,T}).$$

The metrics can be classified into different categories, depending on the following factors.

- **The nature of the metric:** Metrics can be performance indicators, counters, mobile traces, alarms, configuration parameters, and even geographical data. It is important that all the metrics have the same granularity in the time domain. For example, sometimes, the values of the configuration parameters are stored per day, whereas the values of the performance indicators are calculated and stored per hour. In these cases, it is necessary to process the event that indicates a change in the parameter (e.g., the hour at which the change happens) and generate the correspond-

• The type of the cell: The primary metric $\mathbf{m}_p$ is correlated not only with metrics $\mathbf{m}_j$ of the same cell (intracell correlation) but with metrics of the neighboring cells (intercell correlation) as well. In addition, in a heterogeneous infrastructure, the intercell correlation can be classified into different categories, depending on the relation to the primary cell (i.e., the cell under analysis): intrafrequency/interfrequency if the neighboring cell uses the same/different frequency carrier as the primary cell and intertechnological correlation if the neighboring cell uses the same/different RAT as the primary cell. Typically, in the presence of a fault, co-located cells or cells with high overlapped service area have more similar behavior than separate cells. This is a clear advantage in heterogeneous deployments from the diagnosis perspective since the impact of a faulty cell in a neighboring cell is expected to be greater if their coverage areas are more overlapped. To determine which cells are neighbors, the mobile operator can provide a list of adjacencies. If this list is not available, the neighbors can be determined by using handover-related information. In particular, two cells are considered to be relevant neighbors if the number of handovers between them is considerably high. In this sense, a parameter $N_{\text{min HO}}$ is defined to establish the minimum number of handovers per unit time between two cells to be considered as neighbors. In addition, the maximum number of neighbors $N_{\text{neigh}}$ must be also defined. If the maximum allowed is exceeded, then the cells with the lower number of handovers are discarded.

• **The delay/advance of the metric with respect to the primary metric:** The cause-and-effect relationship can lead to some delays between changes in metrics. For example, a change in a configuration parameter may progressively overload the computer processing unit (CPU), resulting in a degradation of the metric CPU congestion one or two samples later. For this reason, each metric can be also shifted one or more samples with respect to $\mathbf{m}_p$. Such a shift can be positive (a delay), zero, or negative (an advance). The effectiveness of this approach relies on the low probability of the occurrence of two faults close in time.

It is worth mentioning that mobile operators use a plurality of metrics that, in some cases, are derived from other metrics or they have a term in common in their formulas. For example, the counter number of drops is included in the definition of the performance indicator retainability so that the same information is reflected by different metrics. The existence of redundancy between metrics may mask more important correlations in the network. Thus, redundancy in the metrics should be avoided as much as possible.

B. Metric Correlation

The first entity of the diagnosis method performs the calculation of a nonweighted correlation between the primary metric
and the rest of the metrics. Formally, the correlation function corr is used to calculate the correlation value \( v_j^{(p)} \) between \( m_p \) and \( m_j \). In principle, the absolute values of two different coefficients have been considered as indicators of correlation, i.e., Pearson’s and Spearman’s coefficients. Their values are in the range 0 to +1, indicating high correlation when the value is close to +1. While Pearson’s coefficient is a measure of the degree of linear dependence between two variables, Spearman’s coefficient measures if the two variables are monotonically related even if their relationship is not linear.

In a simple model, when a problem occurs, there are basically two distinguishable states in the data sequence of a metric, i.e., the interval(s) under the faulty situation and the interval(s) under normal conditions. If the metric is impacted by the fault, the effect in the time series is typically a peak, a step, or something similar to a square/pulse wave representing the two states along the time. Visually, this effect can be more or less evident depending on the metric and its relationship to other factors such as the traffic load. In this simple model, Pearson’s coefficient will be enough to detect the increase or decrease in the metric under faulty conditions with respect to the normal situation.

In a more complex model, we should consider the size of the data sequences to be correlated. If the size is small (as in this paper), the dependence between the observed samples may be nonlinear with greater probability. However, this nonlinearity may only be due to a small number of samples rather than a real nonlinear tendency in the metrics. For this reason, Pearson’s coefficient, which only focuses on linear relationships, is preferred in those cases. In addition, it should be considered that Pearson’s coefficient is susceptible to the effects of outliers, particularly in small data sets. An outlier is a sample that is distant from other samples in the data set. This consideration is important for faults whose duration is short, and its samples can be seen as outliers. In these situations, Pearson’s coefficient will be sensitive to these samples and the fault will be better diagnosed. As shown in Section IV, Pearson’s coefficient appears to be a better choice than Spearman’s coefficient. As a consequence, it has been selected as the correlation coefficient in the proposed method. Considering that \( M \) is the total number of metrics, the computation of the correlation can be expressed as

\[
\begin{align*}
    v_j^{(p)} &= \text{corr}(m_p, m_j) \\
    &= \frac{\sum_{k=1}^{T} m_{p,k} m_{j,k} - T m_p m_j}{\sqrt{\sum_{k=1}^{T} m_{p,k}^2 - T m_p^2} \sqrt{\sum_{k=1}^{T} m_{j,k}^2 - T m_j^2}} \quad \text{(4)}
\end{align*}
\]

where \( T \) is the observation time, and \( m_p \) (and \( m_j \)) is calculated as

\[
    m_p = \frac{1}{T} \sum_{k=1}^{T} m_{p,k} \quad \text{(5)}
\]

In some cases, high correlation values are desirable to be clearly separated from those representing low correlation to provide better diagnosis. Thus, at this stage, a nonlinear function can be introduced in a similar way than activation functions are used in neural networks. A logistic sigmoid function is applied to the correlation values \( v_j^{(p)} \) from (4) as follows:

\[
    v_j^{(p)} = \frac{1}{1 + e^{-l_1 (v_j^{(p)} - l_2)}} \quad \text{(6)}
\]

where \( l_1 \) represents the slope of the curve, and \( l_2 \) is the parameter for horizontal translation. In this paper, suitable values for \( l_1 \) and \( l_2 \) are 5 and 0.75, respectively. Grouping the correlation values, the resulting vector \( v^{(p)} \) is defined as

\[
    v^{(p)} = \left( v_1^{(p)}, \ldots, v_k^{(p)}, \ldots, v_M^{(p)} \right) \quad \text{(7)}
\]

Vector \( v^{(p)} \) is composed of the correlation values of all selected metrics (performance indicators, alarms, etc.) for both the intracell and intercell cases in the order shown previously, i.e., the first elements of the vector correspond to the metrics of the primary cell and the rest of the elements correspond to metrics belonging to neighboring cells. The order of the correlation values of the metrics belonging to the primary cell can be decided by the mobile operator. In addition, the correlation values of the metrics belonging to a neighboring cell should be sorted in the same manner as the metrics belonging to the primary cell.

To cope with delays of metrics with respect to others, shifted versions of a given metric \( j \) should be generated. To explain this, let \( z \) be the maximum magnitude of the time shift; for example, \( z = 2 \) can be assumed. Then, the time shifts \(-2, -1, 0, +1, +2\) lead to metric vectors \( m_{j(-2)}, m_{j(-1)}, m_{j(0)}, m_{j(+1)}, \text{ and } m_{j(+2)} \) and produce the correlation values \( v_j^{(p)}, v_j^{(p)}(1), v_j^{(p)}(2), v_j^{(p)}(3), v_j^{(p)}(4), v_j^{(p)}(5), v_j^{(p)}(6), v_j^{(p)}(7), \text{ and } v_j^{(p)}(8) \), respectively, where the subscript indicates the magnitude and direction of the time shift.

For a time shift equal to \( z \), the following equality is fulfilled (except for border elements in the vector, i.e., those at the beginning or end):

\[
    m_{j,k(z)} = m_{j,k-z(0)}. \quad \text{(8)}
\]

Element \( v_j^{(p)} \) (the one that appears in \( v^{(p)} \); see (7)) is calculated as the maximum value, i.e.,

\[
    v_j^{(p)} = \max \left\{ v_j^{(p)}(k) \right\} \quad \text{for} \quad k = -z, \ldots, 0, \ldots, +z. \quad \text{(9)}
\]

The last step of this stage is to sort the elements of \( v^{(p)} \) according to the average correlation values per cell. This is carried out only for neighboring cells as the metrics of the primary cell are always located at the beginning of the vector. In particular, for any metric \( j \) of each neighboring cell, the following calculation is carried out:

\[
    \text{mean}(v_{\lambda(j)}^{(p)}, \ldots, v_{\lambda(j)+M/(1+\text{NNeigh})}^{(p)}) \quad \text{(10)}
\]

where mean(·) is the average function, \( M/(1+\text{NNeigh}) \) is the number of metrics per cell, and \( \lambda(j) \) is a function that returns
the position of the first metric in $v^{(p)}$ that belongs to the same neighboring cell than metric $j$. In other words, for a given metric $j$, the average correlation of all the metrics belonging to the same neighboring cell is calculated. Thus, the order of the neighboring cells in $v^{(p)}$ is modified according to the obtained values from (10). More specifically, the metrics of the neighboring cell whose average correlation value is the greatest are located immediately after the primary cell. The new vector is denoted by $v^{(p)}$, where

$$\mathbf{v}^{(p)} = \begin{pmatrix} v_1^{(p)}, & v_2^{(p)}, & \ldots, & v_k^{(p)}, & \ldots, & \ldots, & v_M^{(p)} \end{pmatrix}.$$  \quad (11)

### C. Database

The internal database (see Fig. 2) contains the stored fault causes $\mathbf{S}^{(p)}$ and the expert knowledge $\mathbf{E}$. The former comprises information related to past fault occurrences, whereas the latter includes the expert knowledge that will be used to build the weighting factor (as explained later). On the one hand, variable $\mathbf{S}^{(p)}$ is a matrix where each row $s_k^{(p)}$ represents the correlation values for a specific fault cause $k$ given a primary metric $p$. Initially, vector $s_k^{(p)}$ can be formed by only one occurrence of the fault. When a new case of the fault is diagnosed, this vector is updated by the learning entity as explained subsequently. Formally, considering that $F$ is the total number of stored fault causes, matrix $\mathbf{S}^{(p)}$ can be expressed as

$$\mathbf{S}^{(p)} = \begin{pmatrix} s_1^{(p)} \\ s_2^{(p)} \\ \vdots \\ s_F^{(p)} \end{pmatrix} = \begin{pmatrix} s_{1,1}^{(p)} & s_{1,2}^{(p)} \\ s_{2,1}^{(p)} & \cdots & \cdots \\ & \cdots & \cdots & s_{F,M}^{(p)} \end{pmatrix}. \quad (12)$$

Specifically, element $s_{k,j}^{(p)}$ is the correlation value between primary metric $p$ and metric $j$ both with fault cause $k$. The order of the metrics in vector $s_k^{(p)}$ is the same as in (11), i.e., the first metric belongs to the primary cell (intracell case). Since the primary metric can be different for each occurrence of the fault, matrix $\mathbf{S}^{(p)}$ should be defined for each metric $p$ that acts as a primary metric. In general, the number of potential primary metrics is very low as operators typically focus their attention on a few number of performance indicators to detect anomalous behaviors.

On the other hand, variable $\mathbf{E}$ represents the expert knowledge that can be included in the system to support the diagnosis. This variable is a matrix of the same dimensions as $\mathbf{S}^{(p)}$, formally expressed as

$$\mathbf{E} = \begin{pmatrix} e_1 \\ e_2 \\ \vdots \\ e_F \end{pmatrix} = \begin{pmatrix} e_{1,1} & e_{1,2} \\ e_{2,1} & \cdots & \cdots \\ & \cdots & \cdots & e_{F,M} \end{pmatrix}. \quad (13)$$

The operator can use $\mathbf{E}$ as a mask for the weighted correlation, i.e., element $e_{k,j}$ can take values in a certain range [e.g., in the range $(0, 1]$] to give relative importance to metric $j$ in the presence of the fault cause $k$. As a consequence, the operator can give more importance to specific kind of metrics (e.g., performance indicators) or to metrics from a specific cell (e.g., the primary cell). In addition, each vector $e_k$ must be normalized as follows:

$$\sum_{j=1}^M e_{k,j} = 1. \quad (14)$$

### D. Weighting Factor Computation

Once the vector $\mathbf{v}^{(p)}$ is generated, the following step is the calculation of the weighting factor $\mathbf{W}^{(p)}$ that will be used to compute the weighted correlation. As shown in Fig. 2, this factor is derived from two different inputs, i.e., one corresponding to the information provided in the correlation values ($\mathbf{S}^{(p)}$ and $\mathbf{v}^{(p)}$) and one corresponding to the expert knowledge $\mathbf{E}$. This can be mathematically expressed as

$$\mathbf{W}^{(p)} = f \left( \mathbf{C}^{(p)}, \mathbf{E} \right) = \begin{pmatrix} w_1^{(p)} \\ w_2^{(p)} \\ \vdots \\ w_F^{(p)} \end{pmatrix} = \begin{pmatrix} w_{1,1}^{(p)} & w_{1,2}^{(p)} \\ w_{2,1}^{(p)} & \cdots & \cdots \\ & \cdots & \cdots & w_{F,M}^{(p)} \end{pmatrix}. \quad (15)$$

where $\mathbf{C}^{(p)}$ contains the weight or relative importance of metrics based on the information on correlation values, as subsequently defined, and $f(\cdot)$ is the function that combines the two weights $\mathbf{C}^{(p)}$ and $\mathbf{E}$ to produce a new weighting factor that fulfills the following normalization condition:

$$\sum_{j=1}^M w_{k,j}^{(p)} = 1 \quad (16)$$

where $w_{k,j}^{(p)}$ is the weight for the correlation value between primary metric $p$ and metric $j$ with fault $k$.

Matrix $\mathbf{C}^{(p)}$ is used to establish the relative importance of a metric regarding the overall impact of the fault on the respective cell. This matrix is defined as follows:

$$\mathbf{C}^{(p)} = \begin{pmatrix} c_1^{(p)} \\ c_2^{(p)} \\ \vdots \\ c_F^{(p)} \end{pmatrix} = \begin{pmatrix} c_{1,1}^{(p)} & c_{1,2}^{(p)} \\ c_{2,1}^{(p)} & \cdots & \cdots \\ & \cdots & \cdots & c_{F,M}^{(p)} \end{pmatrix}. \quad (17)$$

For a certain cell, the average of the correlation values between the primary metric and the metrics of that cell determines whether the fault has a high impact on the cell or not. Thus, to build matrix $\mathbf{C}^{(p)}$, the average of the correlation values per cell (primary cell included) for both the stored cases $\mathbf{S}^{(p)}$ and
the input case $v^{(p)}$ is first calculated. Then, the minimum value (denoted by min) between these two cases is selected so that the less correlated value establishes the importance given to the correspondent cell. The minimum operator is used to penalize those cells whose correlation to the primary metric is very low. Formally, to calculate the nonnormalized element $c^{(p)}_{k,j}$, the following equation is used:

$$c^{(p)}_{k,j} = \min \left\{ \frac{\sum \left( \frac{u^{(p)}_{\lambda(j)}, \ldots, v^{(p)}_{\lambda(j)} + M/(1+N_{\text{neigh}})}{s^{(p)}_{k,\lambda(j)}, \ldots, s^{(p)}_{k,\lambda(j)} + M/(1+N_{\text{neigh}})} \right)}{1}, \frac{\sum \left( \frac{u^{(p)}_{k,j} + M/(1+N_{\text{neigh}})}{s^{(p)}_{k,j} + M/(1+N_{\text{neigh}})} \right)}{1} \right\}$$  \tag{18}

where $M/(1+N_{\text{neigh}})$ is the number of metrics per cell, and $\lambda(j)$ is a function that returns the position of the first metric in $v^{(p)}$ and $s^{(p)}$ that belongs to the same neighboring cell than metric $j$. As a consequence, the elements from the same cell have the same value. Finally, each vector $c^{(p)}$ of the final matrix $C^{(p)}$ must be normalized, i.e., element $c^{(p)}_{k,j}$ is determined by

$$c^{(p)}_{k,j} = \frac{c^{(p)}_{k,j}}{\sum_{r=1}^{M} c^{(p)}_{k,r}}. \tag{19}$$

With this implementation of matrix $C^{(p)}$, the function $f(\cdot)$ that aggregates $C^{(p)}$ and $E$ to produce $W^{(p)}$ and at the same time fulfills (16) is given by the following expression:

$$w^{(p)}_{k,j} = f \left( c^{(p)}_{k,j}, e_{k,j} \right) = \frac{c^{(p)}_{k,j} e_{k,j}}{\sum_{r=1}^{M} c^{(p)}_{k,r} e_{k,r}}. \tag{20}$$

In a similar way as with matrices $C^{(p)}$ and $E$, other variables could be aggregated to the model to improve the diagnosis. This information, derived from metrics, can be the following examples.

- **Geographical data:** Border cells (i.e., those located on the edge of the network) are less suitable for diagnosis since their metrics are more likely to be degraded. In addition, neighboring cells that are located too far from the primary cell can be penalized by giving them a lower weight.
- **Traffic-related metrics:** Cells with very low traffic should receive minor attention since their metrics may not be statistically significant.

### E. Fault Correlation

The next step of the algorithm is the calculation of the weighted correlation. The most common correlation and distance-based methods could be used here, provided that weights can be easily included in their original definition. In this paper, a variant of Pearson’s correlation coefficient is proposed. In particular, a correlation value is calculated for each stored fault cause $k$, given the primary metric $p$. Thus, for the vector $v^{(p)}$ of correlation values derived from the input, the weight vector $w^{(p)}_{k}$, and the vector $s^{(p)}_{k}$ of correlation values derived from the internal database, the weighted Pearson’s correlation coefficient $r^{(p)}_{k}$ is calculated by using (21), shown at the bottom of the page, where

$$r^{(p)}_{k} = \frac{\sum_{j=1}^{M} w^{(p)}_{k,j} v^{(p)}_{j} - \overline{v^{(p)}_{k}}}{\sqrt{\sum_{j=1}^{M} w^{(p)}_{k,j} v^{(p)}_{j}^2} \sum_{j=1}^{M} w^{(p)}_{k,j} s^{(p)}_{k,j} - \overline{s^{(p)}_{k}}^2}$$ \tag{21}

By calculating this value for the $F$ fault causes, a vector of correlation values is obtained as follows:

$$r^{(p)} = \left( r^{(p)}_{1}, r^{(p)}_{2}, \ldots, r^{(p)}_{F} \right). \tag{24}$$

This vector represents the correlation with each fault cause. As shown in Fig. 2, vector $r^{(p)}$ is the output of this step of the algorithm.

### F. Decision

Let $c$ be the real root cause of the problem (or fault). Then, the final step of the algorithm is the selection of the most likely root cause $c^\prime$; hence, $c = c^\prime$ means that the algorithm has properly identified the cause of the problem. Such a decision is based on two conditions. The first condition is that the value $r^{(p)}_{c^\prime}$ must be the greatest, whereas the second condition is that this value must be above a predefined threshold. Formally, these conditions are expressed as

$$c^\prime = \arg \max_k r^{(p)}_{k} \tag{25}$$

$$r^{(p)}_{c^\prime} > R_{\text{thres}} \tag{26}$$

where $R_{\text{thres}}$ is the threshold that establishes the sensitivity to undiagnosed faults. This means that, if the second condition is not fulfilled, the algorithm is not able to find the cause of the problem, or the previous detection phase provided a false positive (commonly called false alarms).

### G. Learning

The proposed algorithm is able to learn from new fault occurrences. Given an occurrence of the fault $k$ (corresponding to the cause of the problem $c^\prime$ previously identified), the row vector $s^{(p)}_{k}$ must be updated according to some formula or filter
to include this knowledge in the database. The resulting matrix is denoted by $S^{(p)}$. Note that only one row is updated. In this paper, the use of a first-order infinite impulse response filter is proposed. In particular, the expression for each element of the updated vector $s_{k,j}^{(p)}$ is given by

$$s_{k,j}^{(p)} = \alpha \beta_j^{(p)} v_j^{(p)} + \left(1 - \alpha \beta_j^{(p)} \right) s_{k,j}^{(p-1)}$$

(27)

where $s_{k,j}^{(p)}$ refers to the updated value for fault $k$ and metric $j$, $\alpha$ is a user parameter that establishes the importance of past occurrences with respect to the last one, and $\beta_j^{(p)}$ is determined as follows:

$$\beta_j^{(p)} = \text{mean} \left( v_{\lambda(j)}^{(p)}, \ldots, v_{\lambda(j)}^{(p)} + M/(1+N_{\text{Neigh}}) \right)$$

(28)

where $M/(1+N_{\text{Neigh}})$ is the number of metrics per cell, and $\lambda(j)$ is the function previously described that returns the position of the first metric in $v^{(p)}$ that belongs to the same neighboring cell than metric $j$. A high value of $\beta_j^{(p)}$ means that metric $j$ belongs to a cell whose correlation to the problem is significant, and thus, its impact on the matrix of stored cases should be greater.

The update is carried out for all metrics that can act as primary metrics in the diagnosis algorithm. As a consequence, vector $v^{(p)}$ must be calculated not only for the primary metric $p$ derived from the detection phase but also for each potential metric that can be the primary metric in future occurrences of the fault.

IV. PERFORMANCE ANALYSIS

A. Analysis Setup

To show the effectiveness of the proposed method, some examples are provided here. The data of the example are taken from a live LTE network. The selected metrics cover the main aspects of a mobile network (not necessarily LTE) such as accessibility, retainability, coverage, capacity, and mobility. In particular, the following metrics are included in the study.

- **#Drops [# drops events]:** This counts the number of dropped calls in a cell. This metric is essential to determine the existence of problems that directly affect user retainability. In addition, this kind of problems should be treated with the highest priority by troubleshooting experts. Thus, this counter is the primary metric in this example.
- **#Connect [# RRC connection est. attempts]:** This measures the number of connection establishment attempts in a cell at the radio resource control (RRC) layer; thus, it is an estimation of the offered traffic.
- **#Bad_cov [# bad coverage events]:** This collects the number of traffic bad coverage measurement reports received from the UEs. These reports are triggered when the signal level received from the serving cell becomes worse than a threshold. A high value of this metric may be indicative of a coverage hole in that cell.
- **Avg_RSSI [dBm]:** This measures the noise and interference power on the Physical Uplink Control Channel. In cellular networks, there can be radio problems related to interference in the uplink, from either internal or external sources. In such cases, this indicator would be impacted.
- **IRAT_rate [%]:** This calculates the fraction of calls that have been handed over to another RAT with respect to the normal call releases in the same RAT. These kinds of handovers are commonly referred to as inter-RAT handovers. A high value of this metric may reveal, for example, a lack of coverage in a certain RAT.
- **#HO_PP [# handover ping-pong events]:** This counts every handover that is considered to be a ping-pong (i.e., oscillating) handover. The time interval between handovers to trigger a ping-pong event is set to 1 s. A wrong configuration of mobility parameters may lead to successive handovers between the source and target BSs. This problem may significantly decrease the performance of handovers.

The data set is composed of different faults for which the previous metrics are measured. Each fault represents a problem in a certain cell. However, neighboring cells can be also affected by that problem. The selected faults have been verified by troubleshooting experts, and they are common in current networks for any deployed RAT. These faults are as follows:

- **Cell outage (COU):** This is the total loss of radio services in the coverage area of a cell as a consequence of hardware/software failures or other functional faults. Although it can be detected by means of various alarms and alarm correlations, sometimes, those alarms are not accessible (e.g., due to power outages) so that detection is not straightforward. In addition, metrics from the primary (outage) cell cannot be obtained. In this case, a neighboring cell that is affected by the cell outage should be treated as the primary cell.
- **Overload (OVE):** This is given by an eventual sharp increase in offered traffic in a specific area due to casual events such as concerts, football matches, and exposures in shopping centers. As a consequence, many users cannot be served and higher levels of interference can be measured in the network. Although it is not exactly due to a failure, this problem should be addressed in a different way than self-optimization scenarios since a huge increase in traffic may lead to drastic levels of call dropping, which, in turn, can trigger other problems.
- **RRC congestion (RCO):** On certain sites, due to a malfunction of the central processing unit (e.g., a software bug), there can be a progressive increase in RRC failures and a simultaneous increase in RRC re-establishment attempts. As a consequence, the central processing unit will be gradually overloaded until the BS is restarted. Both accessibility and retainability are mainly affected by this problem.
- **External uplink interference (INT):** This is caused by an external source that directly affects the quality of the received signal in the uplink. This problem cannot be treated with conventional self-optimization techniques, where the interference is typically originated from an internal source (i.e., other UEs) due to, for example, a suboptimal configuration of mobility parameters.
For each previous fault cause, three occurrences (or cases) of the fault in different cells have been collected, i.e., 12 occurrences in total. One occurrence per fault will be used to build the self-healing methods (i.e., the training data set), whereas the rest of the cases will be used to evaluate them (i.e., the validation data set). In Fig. 3, the normalized values of the selected metrics during the observation time interval are shown for each of the faults presented in the training data. The faulty situation is represented between the two vertical dotted lines, and its location in time and length is different for each fault cause. This information has been previously determined by troubleshooting experts. Note that the identification of the faulty situation is not needed in the operation of the proposed algorithm. The actual range of the shown metrics is provided in Table I. In general, the ranges greatly vary from cell to cell. In addition, in the same cell, the ranges of the metrics under faulty and healthy situations can be very similar. These are important issues for traditional techniques that analyze the magnitude of the samples rather than other aspects such as the time trend. In most cases, the metrics are also impacted by the traffic pattern, which leads to higher values during the daytime. As observed, the metric #Drops is always degraded as it is the primary metric by which the problem has been previously detected. The impact on other metrics may vary, depending on the specific fault. A cell outage is mainly related to coverage holes, where some characteristic metrics such as #Bad_cov and IRAT_rate are clearly impacted. More specifically, the only metric that is not affected is Avg_RSSI, whereas the metric #HO_PP is only slightly influenced. In the case of overload situations, Avg_RSSI is also affected due to the significant increase in UEs in the area, whereas the decrease in IRAT_rate can be related to congestion in other RATs and/or a high increase in its denominator, which directly depends on the amount of traffic. In the case of RRC congestion, Avg_RSSI, IRAT_rate, and #HO_PP are shown to be poorly correlated to the problem since this fault is not related to interference and mobility issues. Finally, the problem of external uplink interference is mainly given by a pronounced increase in Avg_RSSI that also affects other metrics. On the contrary, metrics such as #Bad_cov and #HO_PP are less affected by this fault.

The following configurations of the proposed algorithm have been tested.

- Basic: This particular implementation only includes metrics from the primary cell (i.e., \( N_{\text{neigh}} = 0 \)) while the learning stage is not supported.
- Basic + Neigh: In addition to the basic configuration, it includes the metrics of one neighboring cell to evaluate whether this additional information coming from other cells improves the diagnosis or not.
- Basic + Learn: This adds the learning stage to the basic configuration. In this case, the training data set (represented by \( S(p) \) and stored in the database) is formed from three occurrences per fault except for the fault cause under evaluation, where two occurrences are aggregated in \( S(p) \), whereas the other is used as input. This aggregation models the learning process of the algorithm after diagnosing a fault.
- Basic + Neigh + Learn: This combines the two previous configurations to take advantage of these features simultaneously.

These configurations are also defined by a set of parameters summarized in Table II, where common and individual settings are detailed. In this paper, the value of \( N_{\text{neigh}} \) is low; thus, \( N_{\text{min \, HO}} \) does not play a key role to discard neighboring cells. In addition, the value of \( \alpha \) is relatively high because, in the beginning, the system should learn faster, whereas the parameter should be progressively reduced as new cases are added.

The proposed system is compared with a fuzzy rule-based system (FRBS) derived from [33] whose fuzzy logic has been adapted to the given problem. With fuzzy logic, the metrics can be interpreted qualitatively, inexactly, or uncertainly. In the context of fault diagnosis, fuzzy logic is a good approach to characterize faults since the expected behavior of the network is typically imprecise when a problem occurs. In such a case, the metrics are treated as linguistic input variables, whose possible values can be low (L) or high (H), depending on whether the metric values are acceptable or not. By a thorough inspection of Fig. 3, an expert could express, by using linguistic terms, which metrics are degraded for each fault. This information, summarized in Table III, represents the knowledge base of the
TABLE I
RANGE OF METRICS FOR EACH FAULT IN THE TRAINING DATA

<table>
<thead>
<tr>
<th>Metric</th>
<th>COU</th>
<th>OVE</th>
<th>RCO</th>
<th>INF</th>
</tr>
</thead>
<tbody>
<tr>
<td>#Drops (# drops events)</td>
<td>0</td>
<td>59</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>#Connect (# est. attempts)</td>
<td>1651</td>
<td>11193</td>
<td>1132</td>
<td>43174</td>
</tr>
<tr>
<td>#Bad_cov (# bad cov. events)</td>
<td>64</td>
<td>1166</td>
<td>77</td>
<td>1461</td>
</tr>
<tr>
<td>Avg_RSSI (dBm)</td>
<td>-114</td>
<td>-110</td>
<td>-116</td>
<td>-105</td>
</tr>
<tr>
<td>IRAT_rate [%]</td>
<td>0.11</td>
<td>2.13</td>
<td>0.43</td>
<td>1.59</td>
</tr>
<tr>
<td>#HO_PP (# ping-pong events)</td>
<td>0</td>
<td>34</td>
<td>7</td>
<td>6914</td>
</tr>
</tbody>
</table>

TABLE II
CONFIGURATION PARAMETERS

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Basic + Neigh</td>
</tr>
<tr>
<td>Meas. Period, L</td>
<td>1 h</td>
</tr>
<tr>
<td>Observ. Time, T</td>
<td></td>
</tr>
<tr>
<td>Max. Time Shift, x</td>
<td></td>
</tr>
<tr>
<td>N_{min,HCO}</td>
<td>0</td>
</tr>
<tr>
<td>N_{neigh}</td>
<td>6</td>
</tr>
<tr>
<td># of Metrics, M</td>
<td>4</td>
</tr>
<tr>
<td># of Stored Faults, F</td>
<td>4</td>
</tr>
<tr>
<td>R_{thresh}</td>
<td>0.4</td>
</tr>
<tr>
<td>Learning Rate, α</td>
<td>0.5</td>
</tr>
</tbody>
</table>

To explain how an input case is diagnosed, the knowledge base of the FRBS must be understood as fuzzy rules following an IF–THEN syntax. An example rule is: IF “metric \( j \) is L” AND “metric \( k \) is H” THEN “fault cause is \( c \).” As observed, the antecedent of the rule is composed of a set of conditions that are combined through the intersection relation denoted by “AND.” Formally, given a time instant \( k \) for a set of \( M \) metrics, the degree of truth \( γ \) of each antecedent is calculated according to the following expression [34]:

\[
γ_{c,k} = \prod_{j=1}^{M} μ_{c,j}(m_{j,k})
\]  

where \( m_{j,k} \) is the metric \( j \) evaluated at the time instant \( k \), and \( μ_{c,j} \) is the membership function for “L” or “H” depending on the fault cause \( c \) and metric \( j \), i.e., row \( c \) and column \( j \) in Table III. Once the degree of truth is calculated for every potential fault cause \( c \) and every sample \( k \) in the time domain, the FRBS chooses the rule whose degree of truth is the greatest. Before this, \( γ_{c,k} \) is averaged in time (denoted as \( \overline{γ}_c \)). Then, the output is computed as

\[
e^{(t)} = \arg \max_c \overline{γ}_c
\]  

where \( e^{(t)} \) is the most likely root cause of the problem. Another difference with the proposed method is that the FRBS must know exactly which are the samples with faults to be diagnosed. On the contrary, in the proposed method, the data sequences used as inputs involve both normal and abnormal samples to calculate the correlation in the time domain.

B. Results

The first experiment is the evaluation of the basic configuration of the proposed algorithm using two different correlation coefficients in the block metric correlation: Pearson’s and Spearman’s coefficients. In addition, the proposed algorithm is compared against the reference method, FRBS. In Fig. 5, vector \( r \) is represented for each evaluated configuration of the proposed method. Each element of this vector corresponds to one of the four fault causes. In the case of the FRBS, the values represented are \( \overline{γ}_c \) for each fault cause \( c \), i.e., \( γ_c \) as a vector. The validation data set is composed of two occurrences per fault cause; hence, eight examples are presented to the methods. The real cause is highlighted in bold type in the figures. As observed, the proposed algorithm using Pearson’s coefficient provides the best performance since the real fault cause receives the greatest value of the fault correlation in all the examples, i.e., its diagnostic success rate is 100%. In the case of
TABLE III
RELATIONSHIP BETWEEN EACH FAULT CAUSE AND METRIC USING FUZZY LOGIC (H: HIGH; L: LOW)

<table>
<thead>
<tr>
<th>Fault</th>
<th>#Drops</th>
<th>#Connect</th>
<th>#Bad_cov</th>
<th>Avg_RSSI</th>
<th>IRA/ rate</th>
<th>#HO_PP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cell outage</td>
<td>H</td>
<td>H</td>
<td>H</td>
<td>L</td>
<td>H</td>
<td>H</td>
</tr>
<tr>
<td>Overload</td>
<td>H</td>
<td>H</td>
<td>H</td>
<td>L</td>
<td>H</td>
<td>L</td>
</tr>
<tr>
<td>RRC congestion</td>
<td>H</td>
<td>H</td>
<td>H</td>
<td>H</td>
<td>L</td>
<td>L</td>
</tr>
<tr>
<td>Uplink interference</td>
<td>H</td>
<td>L</td>
<td>L</td>
<td>H</td>
<td>L</td>
<td>L</td>
</tr>
</tbody>
</table>

Fig. 5. Representation of $r$ and $\gamma$ for the proposed and reference algorithms, respectively. (a) Case 1. (b) Case 2. (c) Case 3. (d) Case 4. (e) Case 5. (f) Case 6. (g) Case 7. (h) Case 8.

using Spearman’s coefficient, this rate is 37.5% (i.e., five cases wrongly diagnosed), whereas in the case of the FRBS, the rate is 75% (i.e., two cases wrongly diagnosed). The main drawback of the FRBS is that it is very dependent on the magnitude of the metric values, whereas the proposed algorithm takes advantage of the abnormal trend over time. Note also that the range of $\gamma$ is typically smaller than $r$ due to the use of the product operator in the related rules. In this sense, only one condition in the antecedent that is not fulfilled would lead $\gamma$ to significantly decrease. However, when all the conditions in the antecedent are fully satisfied for a given fault, the obtained value is notably higher with respect to the others. This situation occurs in cases 7 and 8, as observed in Fig. 5.

A deeper analysis of each fault cause is provided as follows. Regarding the cell outage problem [see Fig. 5(a) and (b)], the most conflicting fault with cell outage is the RRC congestion to the extent that the FRBS selects it as the most probable cause in case 2. The results shown in Fig. 5(c) and (d) are obtained when the overload examples are presented to the methods. In this case, the most conflicting fault is interference since the value of Avg_RSSI is high in both situations, leading to some confusion in the diagnosis. In the case of the proposed algorithm, the values of fault correlation are high in both cases. However, the fault correlation with the real cause is still the highest; hence, the diagnosis is correct. This does not happen with the FRBS in case 3, where the diagnosis is wrong. In Fig. 5(e) and (f), the examples related to the RRC congestion problem are shown. Although the distance with the most conflicting fault is very small, both the proposed method and the FRBS provide a successful diagnosis. Finally, Fig. 5(g) and (h) are related to the evaluation of the interference problem. In the case of the proposed algorithm, the most conflicting fault is overload. More specifically, the conflicts between these two faults are reciprocal according to these results. Regarding the FRBS, it is observed that the real fault cause is orthogonal to the others since the value of $\gamma_{INT}$ is the only nonzero value. As a consequence, the fault is correctly identified in these two cases.

To better explain the big differences between Pearson’s and Spearman’s coefficients, the following example is described. More specifically, Fig. 6 represents the metrics #Drops and Avg_RSSI for the problem of RRC congestion using the training data set. Qualitatively, by looking at the time evolution of these two metrics shown in Fig. 3(c), it can be concluded that the fault clearly impacts on #Drops, whereas the impact on Avg_RSSI is very limited. Thus, to effectively diagnose the problem, the correlation between both metrics is desired to be as low as possible. Numerically, Pearson’s and Spearman’s coefficients are equal to 67% and 86%, respectively, mainly due to their dependence with the traffic load. The value of Spearman’s coefficient is higher because the relationship between #Drops and Avg_RSSI is not linear, as represented by the dashed line.

Fig. 6. Analysis of the dependence between #Drops and Avg_RSSI with the training data set for the problem of RRC congestion.
Fig. 7. Representation of $r$ for different variants of the proposed algorithm. (a) Case 1. (b) Case 2. (c) Case 3. (d) Case 4. (e) Case 5. (f) Case 6. (g) Case 7. (h) Case 8.

in Fig. 6. However, the lower value of Pearson’s coefficient (related to the solid line) is more reasonable for diagnosis purposes since the fault has led to a clear increase in #Drops but not in Avg_RSSI. These results could be also related to the effect of outliers in the correlation coefficients. If the calculations are made without considering the samples under the faulty situation (they are shown within the dotted circle in Fig. 6), the new values are 70% and 81% for Pearson’s and Spearman’s coefficients, respectively. As observed, the presence of outliers produces opposite variations in the correlation coefficients. In the case of Pearson’s coefficient, the presence of outliers in the samples produces a lower value (67%) than if they are not considered (70%). This is in line with the expected results since both metrics have partial correlation due to the traffic load in normal conditions, but there is no clear correlation due to the fault. Thus, it can be concluded that, in the context of this paper, Pearson’s coefficient can provide better results than other coefficients that measure nonlinear relationships.

The next experiment is devoted to the comparison of the proposed algorithm in its different variants. Fig. 7 shows the elements of $r$ for each configuration of the proposed method. As in the previous experiment, the validation data set comprises two examples per fault cause. The name of the real cause is highlighted in bold type. Fig. 7(a) and (b) shows the results when cell outage cases are presented to the system. It is observed that only the solution, including metrics from neighbors and learning phase (the most complete) provides benefits in example (b), where the score for RRC congestion (the most conflicting) is reduced. Such a reduction increases the distance with respect to the score obtained by cell outage (real cause), improving the performance of the algorithm. Fig. 7(c) and (d) represents the values of $r$ when overload situations are presented to the algorithm. In both examples, the Basic + Neigh + Learn configuration reduces the distance between the first and second most probable causes, increasing the probability of a wrong diagnosis. In fact, in example (c), the variant Basic + Learn provides a wrong diagnosis. However, this is the only error produced by the proposed algorithm in any of its configurations. In Fig. 7(e) and (f), the results are related to the RRC congestion problem. More specifically, in example (e), the performance of the configurations, including information from neighbors, is highly improved with respect to the basic configuration since the score of the second most probable cause (overload) is significantly reduced. In example (f), such a reduction is also experienced by the most probable cause; thus, performance is quite similar between variants. Fig. 7(g) and (h) shows the performance of each proposed configuration when interference situations are presented to the system. As in example (e), the configurations, including information from neighbors, achieve better performance due to the distance reduction with respect to the second most probable cause, i.e., the overload problem.

Finally, in Fig. 8, this analysis is summarized by showing the average distance between the real and next most probable fault causes. If the diagnosis is correct, the distance would be given by the two fault causes that obtain the highest values. If not, it would be given by the real fault cause and the most probable fault cause, i.e., a negative value would be obtained in the latter case. Roughly, the larger the distance, the better the diagnosis is expected to be overall. In the figure, it is highlighted that the configuration of the proposed method, including more number of features, i.e., Basic + Neigh + Learn, provides the largest average distance between the first and second most probable causes. As a consequence, better performance in terms of diagnostic success rate should be obtained by using this configuration. In the case of the FRBS, the two wrongly diagnosed cases (i.e., cases 2 and 3 shown in Fig. 5) would lead to negative distances and, as a consequence, to a low average distance.
However, the high distances obtained in cases 7 and 8 significantly improve this figure-of-merit. Regarding the proposed method, it is observed that the configuration that uses data from neighboring cells produces larger average distance than the configuration that implements the learning phase. However, the combination of both features outperforms the basic configuration, even the one that uses data from neighboring cells. These findings are in line with the results described in the previous paragraph.

V. CONCLUSION

Self-healing is increasing its popularity in cellular networks due to the beneficial impact on operator revenues. In addition, as the number of network elements and devices is exponentially growing, the intelligent use of network measurements becomes a key feature for automatic troubleshooting. This paper has proposed a self-healing algorithm that analyzes time series of cell metrics under problematic situations to determine the fault cause. In comparison to the state of the art, the proposed method takes into account the time dependence of network metrics and the impact of the fault on neighboring cells to achieve better diagnosis accuracy. The proposed method is designed for current and future deployments in the RAN, where a vast plurality of metrics related to the affected cell and surroundings may be measured. The correlation between these metrics is computed to characterize the state of the network. Then, this state is compared with stored faulty patterns through the calculation of a weighted correlation to provide the diagnosis. This correlation is modulated by effective weights that are built according to the previously calculated metric correlation and expert knowledge, if required.

Evaluation of the algorithm has been carried out by using a set of fault cases collected from a live network. In particular, representative metrics from LTE cells have been selected for the analysis. The proposed algorithm has been compared to a reference system based on fuzzy logic, which is a technique commonly used in self-healing to identify faults. Results show that the metric ranges greatly vary between cells. This is an important drawback for the reference method, which relies on the magnitude of the metrics to discover patterns. Conversely, the proposed algorithm takes advantage of the time trends to identify faulty behaviors, obtaining significantly better diagnostic success rate. Finally, different variants of the proposed algorithm have been assessed. It has been shown that adding information about the surrounding area (e.g., from neighboring cells) together with learning from past occurrences of the same fault improves the proposed algorithm.

From a practical perspective, it is crucial for operators to maintain a complete database where each problem is saved along with its diagnosis. In many situations, the fault is solved without knowing (or registering) the real cause. Other important challenge is the selection of the number of metrics and the number of neighboring cells. Although these two variables are related to the number of faults that could be identified, an excessive value of these variables might not lead to higher accuracy of the system and would involve higher computational cost.


Pablo Muñoz received the M.Sc. and Ph.D. degrees in telecommunication engineering from the University of Málaga, Málaga, Spain, in 2008 and 2013, respectively.

From 2009 to 2013, he was a Ph.D. Fellow in self-optimization of mobile radio access networks and radio resource management. Upon completing the Ph.D. degree, he has continued his career with the University of Málaga as a Research Assistant within an R&D contract with Optimi-Ericsson focusing on self-organizing networks. In 2014, he held a postdoctoral position granted by the Andalusian Government in support of research and teaching.

Isabel de la Bandera received the M.Sc. degree in telecommunication engineering in 2009 from the University of Málaga, Málaga, Spain, where she is currently working toward the Ph.D. degree in telecommunications engineering.

In 2008, she worked on radio-frequency identification projects with the Department of Communications Engineering, University of Málaga. Since February 2010, she has been working on projects concerning radio resource management in next-generation mobile networks.

Emil J. Khatib received the M.Sc. degree in telecommunications engineering in 2010 from the University of Málaga, Málaga, Spain, where he has been working toward the Ph.D. degree with the Department of Communications Engineering since spring 2012.

Until 2012, he was a Research Assistant with the Department of Systems Engineering and Automation, University of Málaga. His research interests include communication protocols and artificial intelligence applied to telecommunications.

Ana Gómez-Andrades received the M.Sc. degree in telecommunication engineering in 2012 from the University of Málaga, Málaga, Spain, where she is currently working toward the Ph.D. degree in self-healing Long-Term Evolution networks.

She is currently with the Department of Communications Engineering, University of Málaga, in cooperation with Ericsson. Her research interests include mobile communications and big-data analytics applied to self-organizing networks.

Inmaculada Serrano received the M.Sc. degree from the Universitat Politècnica de València, València, Spain. She collaborated with the Department of Communications, investigating arrays of antennas and completed her research with the University of Sheffield, Sheffield, U.K., in 2003. She specialized further in radio after receiving the Master’s degree in mobile communications from the Universidad Politécnica de Madrid, Madrid, Spain.

In 2004, she joined Optimi and started a broad career in optimization and troubleshooting of mobile networks for different operators around the world, including a variety of consulting, training, and technical project management roles. In 2012, she moved to the Advanced Research Department, Ericsson, Málaga, Spain, with a strong focus on automatic troubleshooting and self-healing.

Raquel Barco received the M.Sc. and Ph.D. degrees in telecommunication engineering from the University of Málaga, Málaga, Spain, in 1997 and 2007, respectively.

She was with Telefónica, Madrid, Spain, and with the European Space Agency, Darmstadt, Germany. From 2000 to 2003, she worked part-time for Nokia Networks. In 2000, she joined the University of Málaga, where she is currently an Associate Professor. Her research interests include satellite and mobile communications, mainly focusing on self-organizing networks.