Self-healing framework for next-generation networks through dimensionality reduction

David Palacios, Sergio Fortes, Isabel de-la-Bandera and Raquel Barco

***Abstract*—Next generation self-organizing networks (NG-SONs) are the key that will lead to the full automation of the network management in the forthcoming generations of cellular communications. New challenges, like the deployment of novel wireless services or the aim of operators at providing an end-to-end monitoring and optimization, make it necessary to develop an innovative scheme for network management. In this paper, a self-healing (SH) framework for next-generation networks using dimensionality reduction is proposed as the tool enabling the management of an increasingly complex network, taking advantage of both feature selection and feature extraction techniques. A proof of concept has been carried out in the context of automatic diagnosis in a live network. Results show that the proposed framework can effectively manage a high-dimensional environment, eventually automating the tasks usually performed by troubleshooting experts while optimizing the performance of the diagnosis tool.**

***Index Terms*—Self-organizing networks (SONs), self-healing, dimensionality reduction, feature selection, feature extraction, fault diagnosis, root cause analysis (RCA), performance indicator.**

I. INTRODUCTION

FUTURE cellular networks are expected to present a level of complexity never seen before in their way toward 5G [1]. Thus, the idea of networks aiming at a fully automated and optimized management arises as a pressing need. This is the goal of NG-SONs, which, as of today, support operators in their daily tasks, contributing to decrease both the operational expenditure (OPEX) and the capital expenditure (CAPEX) [2].

SONs, and especially, the field of self-healing (SH), is built on the concept of network performance indicators. These provide operators with information about the current state of the network, allowing performance degradations to be detected and their eventual root cause to be diagnosed and repaired. Currently, network performance indicators range from alarms and event counters to more complex metrics. However, with the advent of 5G, new sources of information are expected to be added to the former. In order to have a broader view beyond the network itself, context information is planned to be included in NG-SON mechanisms [3]. In addition, end-to-end performance indicators including service-specific metrics are intended to be assessed in new cellular networks as a means of enhancing the quality of experience (QoE) perceived by the user.

This scenario leads to an increase in the already huge amount of sources of information in cellular networks, some of which may not be truly relevant for the network management. In particular, this volume and variety of available data impacts on two aspects regarding the network management. On the one hand, it poses a storage problem: large databases in the Operations Support System (OSS) must be deployed to collect a continuous stream of information coming from different monitoring processes all along the network. On the other hand, network experts are in charge of identifying the most relevant performance indicators for each SON use case, so that the resulting SON functions can be simple enough to be easily handled, but sufficiently complex to manage some aspects of the network behavior.

In the context of automatic diagnosis in SH, troubleshooting experts would have to select those indicators which best allow a performance degradation to be detected and a given fault cause to be diagnosed. This selection has been traditionally performed by hand, according to the operators' expertise. Examples of this can be found in recent works on self healing networks [4], [5]. Given the ever increasing number of network functionalities, the variety of indicators related to these and the diversity of issues that may consequently arise, this selection may often lead to sub-optimal SH functions due to the human bias. Furthermore, this selection is often a highly time-consuming task, given the not always evident relation between indicators and network faults. Therefore, a tool for the automatic selection of relevant performance indicators or the computation of a reduced set of new indicators in cellular networks is eagerly needed. That is, a tool for dimensionality reduction in which the performance indicators are the dimensions (also referred to as features) on which to act.

On this line, some steps have already been taken in an attempt to find a small subset of performance indicators leading to a minimum diagnosis error rate (DER) of a subsequent diagnosis algorithm. In [6], a genetic algorithm is used to this purpose. In [7], the selection of performance indicators is carried out using an unsupervised technique, allowing the application of feature selection over databases of unlabeled data.

Apart from self healing in cellular networks, there are other fields that have come across with the same problem, and that, consequently, have used tools for dimensionality reduction. In the field of computer vision, an image may be seen as a heavy container of scarce useful information and a lot of useless data [8]. In medicine, it is often needed to identify and isolate the subset of genes behind a certain pathology [9].

In this paper, first, feature selection and feature extraction techniques are presented, together with their main benefits and drawbacks in the field of automatic diagnosis in self-

The authors are with the Andalucía Tech., Departamento de Ingeniería de Comunicaciones, Universidad de Málaga, 29071 Málaga, Spain (e-mail: dpc@ic.uma.es; sfr@ic.uma.es; ibanderac@ic.uma.es; rbarco@uma.es).

healing. Then, a framework for automatic diagnosis in self-healing incorporating both kinds of techniques is proposed, beyond the works in which dimensionality reduction is only addressed by feature selection ([6], [7]). This framework allows using both labeled and unlabeled data, as well as integrating performance information from different sources. Furthermore, it supports and even relieves troubleshooting experts from dealing with such amount of different data (decreasing the OPEX). Regarding the OSS, it enables a reduction of the network storage needs, as well as the eventual complexity of the self-healing mechanisms (decreasing the CAPEX). Finally, a proof of concept is carried out comparing different techniques for dimensionality reduction using data gathered from a live cellular network.

II. Feature selection

Within dimensionality reduction, feature selection consists in identifying the subset of features which are considered as the most relevant according to a certain criterion or target. It is usual that this kind of techniques are implemented as a feature scoring algorithm with a given threshold. In this way, a feature selection technique acts as a filter: only those attributes above the threshold pass through the filter, whereas the others are discarded and considered as non relevant.

Besides, within feature selection techniques, there is a wide variety of both supervised and unsupervised techniques. The first use the information contained in the features as well as a *ground truth* label. In a classification or diagnosis problem, this label describes the class that a given sample belongs to. In SH, this label stands for the network state under which that sample was collected. Supervised methods try to find the features whose behavior varies the most depending on these labels (e.g., [6]). Conversely, unsupervised methods only rely on the information contained in the features themselves. These techniques are used when no labeled samples are available. However, given a set of labeled samples, the usage of supervised or unsupervised techniques may depend on different criteria. Although in this case, supervised techniques are often preferred, unsupervised techniques (e.g., [7]) may allow finding features which reveal underlying classes that were not initially considered among the set of *ground truth* labels.

A schematic for a feature selection technique is shown in Figure 1a. This technique takes as its input the pair $(\bar{x}, y)$, where $\bar{x}$ stands for a vector of N features ($\bar{x} = \{x_1, x_2, \ldots, x_N\} \in \mathbb{R}^N$) and y stands for the *ground truth* label, which may be present (to be used by a supervised technique) or not. As a result, a list of selected KPIs is provided and used to filter $\bar{x}$, leading to $\hat{x}$. Now, $\hat{x} = \{x_i, \ldots x_j\} \in \mathbb{R}^S$, with $i, j \in \{1, \ldots, N\}$, $i \neq j$ and $S < N$.

The fact that the selected features make up a subset of the original feature set brings a big advantage from the point of view of human experts: the resulting performance indicators preserve the meaning of those at the input. This way, they are still comprehensible by troubleshooting experts. Related to this, automatic selection of performance indicators entails a second advantage. If the most relevant performance indicators for a certain purpose are known in advance, then the monitoring of all the indicators considered as non relevant might be disabled. This would lead to a significant reduction in the storage needs of the network databases, even though the selection should be reassessed every certain time.

III. Feature extraction

Unlike the selection approach, in feature extraction a new set of features is built from the original feature set, so that the number of the resulting new features is lower than the number of the original ones. From the mathematical point of view, feature extraction is a tool that allows projecting the original features onto a more convenient and reduced basis. New features are built in such a way that they retain as much information as possible from the original feature set. In the context of cellular networks this means that a new set of synthetic performance indicators are built upon the combination of the original ones.

Feature extraction techniques may be classified into linear and non-linear techniques. The first group is made up of techniques whose output are weights to be directly applied to the original feature set, so that the new features result from a linear combination of the first. One of the best known and most widely used technique of this kind due to its simplicity and effectiveness is principal component analysis (PCA). This technique performs an orthogonal transformation to convert a set of observations of possibly correlated features into a set of uncorrelated variables called principal components. In non-linear feature extraction techniques, however, a non-linear transformation is applied to the original feature set. Kernel PCA (kPCA) or self-organizing maps (SOM) are examples of widely used non-linear feature extraction techniques.

Feature extraction poses an interesting advantage when compared to feature selection. Given that the new synthetic features are computed based on the combination of those from the original feature set, such indicators could contain a much higher amount of useful information than if a technique for feature selection was used. That is, the original feature set could be compressed into a smaller set of rich information synthetic performance indicators. This is specially useful for many machine learning techniques, which are commonly used as SON functions, and which usually suffer from issues related to the sparsity of data along the features they use.

Feature extraction, however, poses some non negligible drawbacks. First, the synthetic performance indicators generated this way are not comprehensible by troubleshooting experts, making it difficult to relate them to a given network state. And second, every time a new sample for a synthetic performance indicator is to be computed, a value for all of its components is needed, needing all of them to be permanently monitored.

A schematic for feature extraction is shown in Figure 1b. In this case, given that the vast majority of feature extraction techniques is unsupervised, only $\bar{x}$ is used as its input. The result is a model for KPI transformation. This model may differ depending on the particular technique being used. For example, it may consist of a matrix describing how the

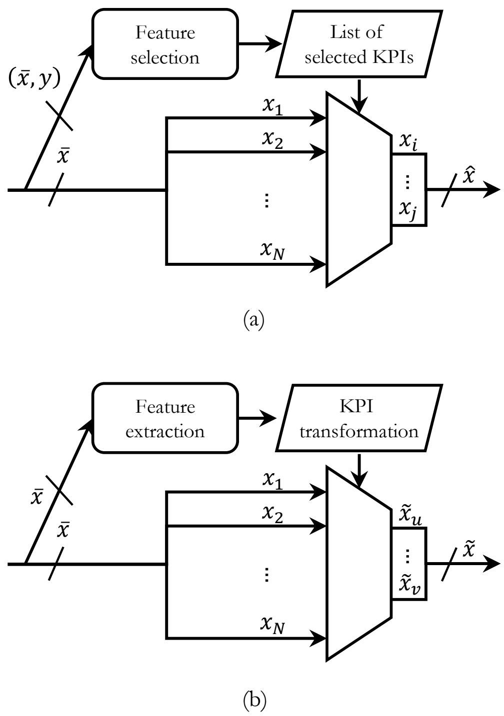

Fig. 1. Schematics for feature selection (a) and feature extraction (b) in the context of self-healing functions.

indicators at the input are linearly combined to produce the synthetic ones. It may also describe non-linear functions to be applied over the former to get the latter. In any case, the resulting output takes the form of $\tilde{x} = \{\tilde{x}_u, \ldots, \tilde{x}_v\} \in \mathbb{R}^E$, with $E < N$, where the tilde highlights the fact that the features at the output are different from those at the input.

IV. Dimensionality reduction-based self-healing framework

SH functions often consist of three phases: a design phase, usually involving a network expert (referred to as troubleshooting expert), followed by a training phase and an operating phase, in which SH functions operate autonomously. The design phase is in which troubleshooting experts apply their knowledge. First, they are in charge of selecting the set of network states to be eventually identified and repaired. Next, they select the performance indicators that, to their knowledge, best allow those states to be eventually diagnosed. And finally, they are in charge of providing a knowledge base for SH functions, usually in the shape of a set of samples, labeled by them after having been analyzed. Each label stands for the network state that the troubleshooting expert infers regarding the values of the performance indicators. Thus, each sample may be seen as an N+1-dimensional vector, where the first N components stand for the performance indicators used to monitor the network status, and the remaining component stands for the label. In the training stage, SH functions learn how network states and performance indicators relate to each other. To that end, a system model is built by analyzing the collection of labeled network samples provided by the troubleshooting expert. Finally, during the operating phase, SH functions predict a network state and even decide which actions to take given a set of new unlabeled samples and the system model from the training phase.

Figure 2 shows the proposed framework for SH functions, in which the upper block covers the design and training phases, and the lower block covers the operating one. These blocks are further divided into two sections by a dotted vertical line. The left side (which is common in both the upper and lower block) corresponds to a data acquisition and formatting stage. It is in charge of gathering and integrating data coming from different sources. The right side is the one performing the dimensionality reduction and data forwarding to the SH function.

The main target of the data acquisition and formatting stage is to provide SH functions with information as much varied and detailed as possible: from the RAN- (radio access network) and core-level information that is used in current SON deployments, to UE- and context-level information. As of today, cell-level performance indicators may be directly found in the OSS databases, in the form of event counters or more elaborated metrics, computed from the first. The information from the UE may be retrieved in two different ways: in a network-managed way, following the minimization of drive tests (MDT) functionality [10], and by means of specific software running on the UE, acting as an end-user probe. In the first case, this information is stored in base stations as UE logs, consisting of signaling trace records. For example, a register of call trace events collected along the Uu, S1 and X2 interfaces and radio link power and quality measurements for LTE (long-term evolution) networks. In the second case, these probes allow operators to retrieve service-specific end-to-end performance metrics, providing information about the quality of service (QoS) experienced by end-users. Regarding context information, the most used type is user location and speed, which can be retrieved from MDT logs. Other sources of context information are social networks and over-the-top (OTT) applications. In these cases, network operators and service providers must reach an agreement, so that the former can retrieve the information collected by the latter. Finally, in line with the end-user perceived quality of experience (QoE), information from the users' complaints may be retrieved using a customer experience management (CEM) tool.

Having the information from different sources collected, a formatting process is needed for the resulting indicators to be expressed in an homogeneous format. On the one hand, this procedure includes making these indicators share the same temporal resolution. And, on the other hand, it includes transforming them into quantitative variables in case they were not. For example, in the case of the customers' complaints. As a result, a sample made up of N indicators may be expressed as $\bar{x} = \{x_1, x_2, \ldots, x_N\}$ (see Figure 2, both in the upper and lower blocks).

Regarding the dimensionality reduction section, a different approach is followed in each block in Figure 2. Starting with the upper block (design and training phases), a feature selection technique is proposed to be used over $\bar{x}$. This sets the troubleshooting expert aside from having to analyze and select the most relevant performance indicators for the SH

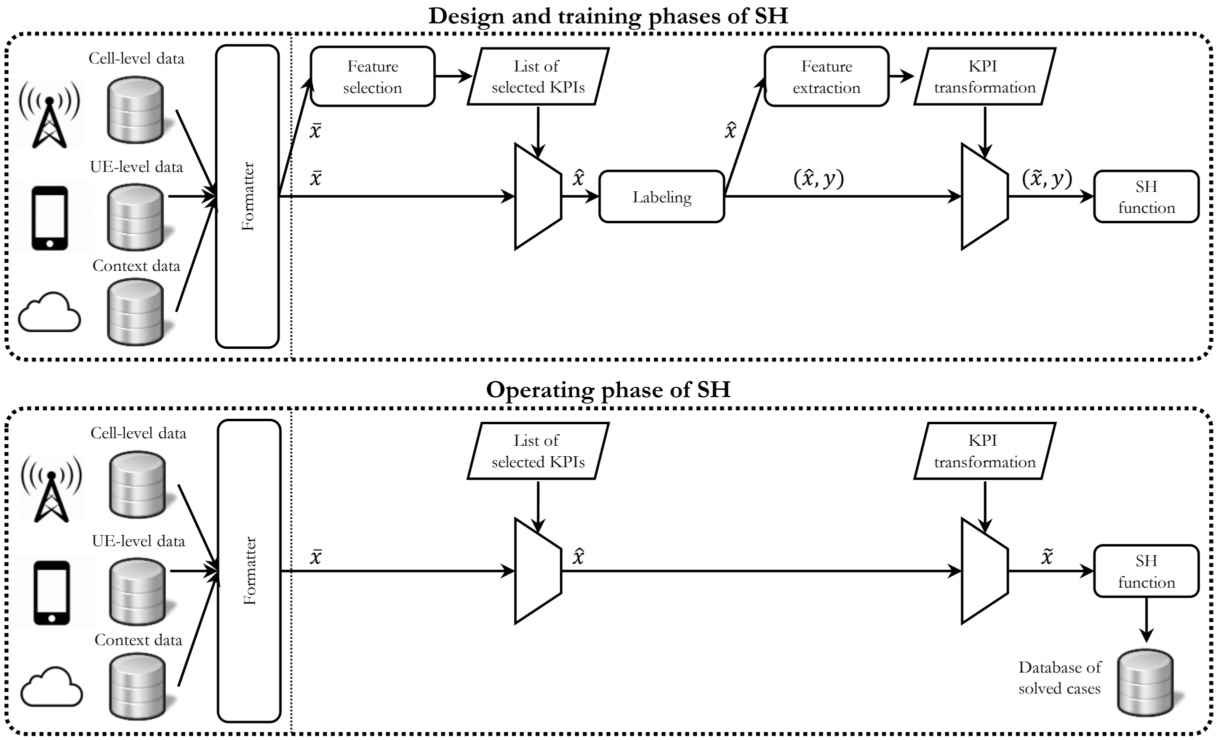

Fig. 2. Proposed framework for next-generation self-healing networks, including different data sources and dimensionality reduction techniques.

function. At this point, and given that no label is attached to the samples, an unsupervised technique for feature selection should be used (e.g., [7]). This will result in a list of selected key performance indicators (KPIs), which will be used to filter $\bar{x}$, resulting in $\hat{x} \in \mathbb{R}^S$, with $S < N$. The next step is to provide a network state label, y, to each sample, resulting in the pair $(\hat{x}, y)$. Despite the aim at minimizing human intervention, it is necessary to use troubleshooting experts' knowledge in early network deployments for this task. However, assuming a sufficiently mature network, meaning the availability of a populated enough database of past solved cases, the troubleshooting expert could be relieved of analyzing new cases for new re-trainings. Having such database, these labels could be provided by a case-based reasoning (CBR) tool. This tool would verify how similar the new cases under evaluation are with respect to those that have already been evaluated in the past by the SH function. That would also be useful for the feature selection technique, which could be supervised instead.

Next, in order to optimize the way how information is provided to the SH functions, the usage of a feature extraction technique is proposed. It intends to further reduce the number of performance indicators at the input of SH functions, in an attempt of compacting the indicators already selected into a reduced number of new synthetic indicators. This allows SH functions to be more time efficient and less prone to overfit, one of the most common issues in machine learning. As a result, a model for KPI transformation is provided. This transformation is applied over the pair $(\hat{x}, y)$ (only affecting $\hat{x}$), resulting in the pair $(\tilde{x}, y)$, which is used for the training of the SH function. Now, $\tilde{x} \in \mathbb{R}^E$, with $E < S$.

Once that the SH function has been trained (i.e., that the model relating $\tilde{x}$ and y has been built), it can be used in the operating phase. In this phase, the new unlabeled samples to be evaluated, $\bar{x}$, must be first reduced to the form $\tilde{x}$. Thus, a filter using the list of selected KPIs derived in the prior phases is used, without having to compute the list again. In order to reduce the storage needs of the network databases, this filter could be implemented by only monitoring and storing the performance indicators in the list. In the same way, the KPI transformation derived in prior phases is applied next to get $\tilde{x}$, without having to compute it again, feeding the SH function. Finally, the resulting output of this function is stored in a database of past cases, so that it helps automating future re-designs and re-trainings. At this point, it is the operator's decision whether to store $\bar{x}$, $\hat{x}$ or $\tilde{x}$ together with the resulting output, being a trade-off between the capability to improve future feature selection and labeling and their storage requirements, respectively.

V. Proof of Concept

A proof of concept has been carried out to assess the proposed framework. To this purpose, a diagnosis tool consisting in a linear discriminant analysis (LDA) classifier has been used. Due to the unavailability of a dataset from a live network simultaneously containing cell-, UE- and context-data, this proof of concept has been divided into two tests. In the first, a high-dimensional dataset ([11]) only containing cell-level data from a live network is used to prove the benefits of the proposed framework in terms of dimensionality reduction. In the second, a medium-dimensional simulation-based scenario ([12]) containing cell-, UE- and context-data is used to prove its benefits in terms of data integration.

A. Test 1: dimensionality reduction

1) Experiment setup: A 359-sample dataset from the OSS of a live LTE RAN has been used. The samples are made up of 286 cell-level performance indicators (ranging from RRC event counters, to CPU and volume metrics) plus a *ground truth* label. These labels were previously elicited from troubleshooting experts and correspond to four different abnormal situations: *high traffic*, *no traffic*, *high CPU utilization* and *low coverage*.

In this test, 30% of the samples are used by the dimensionality reduction techniques to compute the list of selected KPIs and their subsequent transformation (see Figure 2). Another 50% are used in the training phase of the SH function, and the remaining 20% are used to test the performance of the classifier in terms of its DER. This test is repeated 100 times, given the small sample size, shuffling the samples belonging to each of these three subsets in each repetition.

Seven different situations are assessed in order to test different schemes for dimensionality reduction in SH. The first situation, shown as a baseline, results from taking all the available indicators from the database and using them as the input for the diagnosis algorithm. In the second case, a troubleshooting expert (abbreviated as TE onwards) is asked to select the subset of indicators that, to his knowledge, better represent the variety of underlying fault causes. In this case, 20 out of the 286 available indicators were selected. Next, two different unsupervised techniques for feature selection have been used [13], [7] (abbreviated as U1 and U2). This represents the case of an early deployment of a cellular network, when no knowledge from past cases is available. U1 selects the features according to the so-called Laplacian score. U2 uses a clustering stage followed by a supervised technique from feature selection to make the overall method unsupervised. Following, there are two cases of dimensionality reduction through supervised techniques for feature selection (abbreviated as S1 and S2 onwards), corresponding to [14] and [15]. These cases correspond to a situation in which a certain amount of past solved cases is available. In [14], a sequential feature selection technique is used, whereas in [15], technique known as neighborhood component feature selection is used. In U1, U2, S1 and S2, 20 indicators have been selected, so the comparison among these and TE can be fair. Finally, the proposed framework (FR) has been tested. To that end, S2 has been used, followed by a PCA module for feature extraction. In this case, we took the first four principal components, retaining 90% (on average) of the total variance of the indicators selected by S2 (Figure 3).

An additional test is carried out to show the trade-off between the number of synthetic indicators provided by the feature extraction module and the resulting performance of the SH tool. To that end, a PCA module is used following S2 using an increasing number of considered principal components and assessing the subsequent DER.

2) Results and discussion: In Figure 3, each box plot shows the first, second and third quartile as well as the lower and upper adjacent values for each of the situations for dimensionality reduction throughout the 100 iterations. Outliers are shown as crosses. As it can be seen, the worst values for DER correspond to the usage of all the indicators, since many of them may contribute as noise sources in the diagnosis process. The selection of TE shows a better performance compared to that of the *all indicators* case, at the expense of having spent a long time identifying which indicators could be more related to the underlying fault causes. From here on out, all the cases for dimensionality reduction show their capabilities to relieve the TE from making this selection, showing S2 and U2 as the best techniques for supervised and unsupervised feature selection, respectively. Following any of these approaches, the management costs would be reduced by a decrease of a 93% (20 out of 286 indicators) the volume of the OSS databases. Furthermore, FR achieves the minimum median DER while only using four indicators. That is, further achieving a reduction of 98.6% the databases size.

Table I shows 10 out of the 20 cell-level performance indicators that TE selected (upper side). The lower side of the table shows some of the indicators that S2 selected. As it can be seen, the KPIs that TE selected tend to be magnitudes more comprehensible or interpretable to a human, like the handover success rate or the accessibility. However, these might not be the best to identify a given set of network faults. S2 avoids this human bias and, instead, selects a set of indicators that despite being less directly interpretable by a human expert,

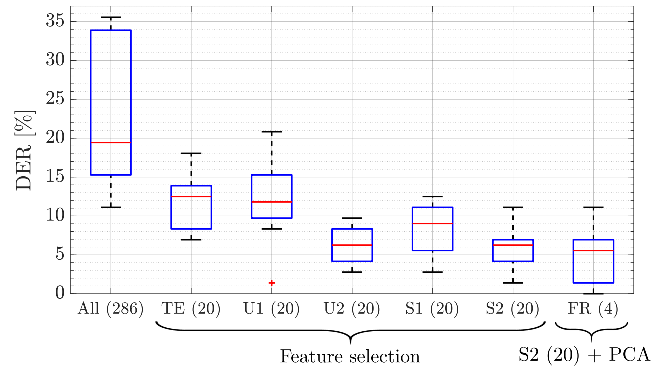

Fig. 3. DERs for an LDA classifier given different methods for dimensionality reduction over performance indicators. TE = Troubleshooting Expert; U1 [13]; U2 [7]; S1 [14]; S2 [15] and FR (the proposed framework). The number of indicators used as the input of the SH function is shown between parentheses.

TABLE I
SELECTION OF KPIS

Troubleshooting expert	
1. Dropped call rate	2. Average CQI
3. CPU load 60%-80%	4. Retainability
5. E-RAB estab. succ. rate	6. Handover succ. rate
7. Uplink data volume	8. Downlink data volume
9. #Bad cov. eval. rep.	10. Accessibility
Supervised feature selection technique S2 ([15])	
1. #-2 $<$ $SINR_{PUSCH}$ $\leq$ 2 dB	2. #10% $<$ PRB DL utiliz. $\leq$ 20%
3. #Bad cov. eval. rep.	4. #-9 $<$ $SINR_{PUCCH}$ $\leq$ -6 dB
5. #HARQ failure UL QPSK	6. Traffic volume PDCP DRB DL
7. UE average session time	8. #RRC conn. establ. succ.
9. #RRC conn. estab. att. MOS	10. #(0 $<$ UE UL throughput (PDCP SDU) $\leq$ 1 Mbps)

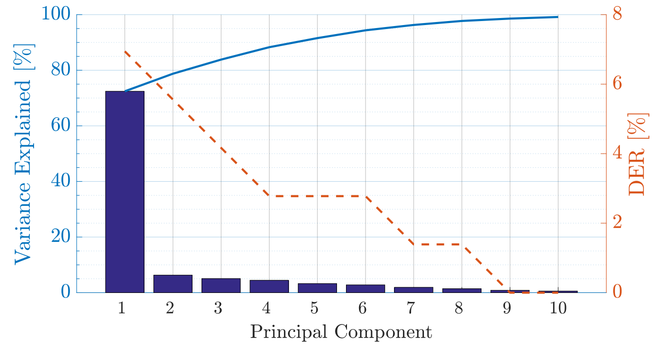

Fig. 4. On the left axis: individual (bars) and cumulated (solid line) percentage of the variance explained by the first to the tenth principal component of PCA. On the right axis, dashed line: resulting DER from taking an increasing number of principal components as the performance indicators at the input of the diagnosis SH function.

carries the kind of information needed for the purpose. For example, several specific histogram-like counters are selected, like the number of times that the DL PRB (physical resource block) utilization ranges from 10% to 20%, rather than less specific indicators, like the DL data volume.

Figure 4 shows the results (averaged over the 100 iterations) when the synthetic performance indicators generated at the output of the feature extraction technique are used at the input of the diagnosis tool. The left axis shows in blue bars in descending order the explained variance of the synthetic performance indicators (i.e., the principal components). The explained variance is a ratio: the variance calculated on a given subset of features with respect to the variance calculated on the whole feature set. The cumulated explained variance is shown in a blue solid line on the same axis. The right axis shows the corresponding subsequent DER. Nine synthetic performance indicators are enough to reduce the DER to a 0%.

B. Test 2: data integration

1) Experiment setup: This simulation-based dataset contains 574 one-minute cell-level samples and 574×600 UE-level 0.1-second samples. The states *overshoot*, *interference*, *weak coverage* and *normal state* are forced on a cell, gathering its indicators (e.g., call drop rate or PRB utilization) and those from two of its neighbors. Only the UEs served by these cells have been monitored. As for them, measurements over the radiolink (RSRP, RSRQ, SINR) and throughput make up UE-level data, including their position as context information. This leads to a 78-dimensional dataset when all these features are considered jointly.

Three different situations for data integration for diagnosis have been assessed: using only cell-level data, using only UE-data, and using both together. Each situation has been further divided in two (leading to six situations), applying or not the proposed framework over the integrated data. In this test, the formatting stage (Fig. 2) consists in averaging 600 samples of UE data each minute. For each situation, the same split, shuffling and number of repetitions as in Test 1 have been performed. In this test, FR consists in S2 selecting 20 indicators followed by PCA taking as many component carriers as to gather 95% of the explained variance (five of them, on average).

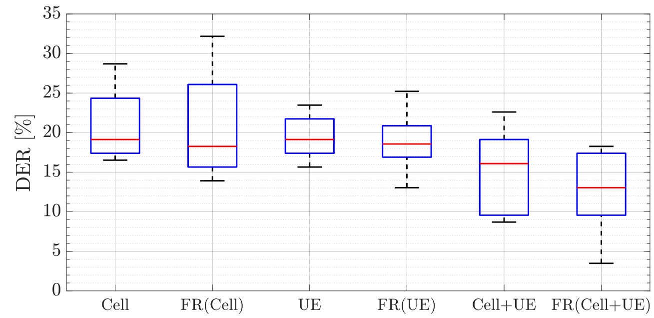

Fig. 5. DERs of the diagnosis tool given different sources of data and integration methodologies.

2) Results and discussion: Figure 5 shows that the best case, FR(Cell+UE), corresponds to applying the proposed framework over the integrated data sources. In such case, the diagnosis tool benefits both from the richness of the data provided and from their low dimensionality, two conditions that are not present in any of the other situations.

VI. Conclusion

In this work, for the first time, a scheme for dimensionality reduction made up of feature selection and feature extraction techniques has been proposed in the context of cellular networks. This scheme enables a self-healing framework for next-generation networks, capable of integrating different data sources. Taking advantage of the benefits of both kinds of dimensionality reduction approaches, the proposed framework has proven its capability to relieve a network expert of analyzing and selecting the most relevant performance indicators in SH tasks and to reduce the storage needs of the OSS databases while optimizing the performance of these tasks.

Acknowledgment

This work has been partially funded by Junta de Andalucía (Proyecto de Investigación de Excelencia P12-TIC-2905) and the Spanish Ministry of Economy and Competitiveness (TEC2015-69982-R).

References

[1] F. Boccardi, R. W. Heath, A. Lozano, T. L. Marzetta, and P. Popovski, "Five disruptive technology directions for 5G," *IEEE Communications Magazine*, vol. 52, no. 2, pp. 74–80, February 2014.

[2] 3GPP, "Self-Organizing Networks (SON); Concepts and requirements, version 14.0.0 (2017-04)," TS 32.500.

[3] S. Fortes, A. A. Garcia, J. A. Fernandez-Luque, A. Garrido, and R. Barco, "Context-aware self-healing: User equipment as the main source of information for small-cell indoor networks," *IEEE Vehicular Technology Magazine*, vol. 11, no. 1, pp. 76–85, March 2016.

[4] A. Gómez-Andrades, P. Muñoz, E. J. Khatib, I. de-la Bandera, I. Serrano, and R. Barco, "Methodology for the design and evaluation of self-healing LTE networks," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 8, pp. 6468–6486, Aug 2016.

[5] D. Palacios, E. J. Khatib, and R. Barco, "Combination of multiple diagnosis systems in self-healing networks," *Expert Systems with Applications*, vol. 64, pp. 56 – 68, 2016.

[6] M. Kajó and S. Nováczki, "A genetic feature selection algorithm for anomaly classification in mobile networks," in *19th International ICIN conference - Innovations in Clouds, Internet and Networks*, Mar. 2016.

[7] D. Palacios and R. Barco, "Unsupervised technique for automatic selection of performance indicators in self-organizing networks," *IEEE Communications Letters*, vol. 21, no. 10, pp. 2198–2201, Oct 2017.

[8] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. fisherfaces: recognition using class specific linear projection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 711–720, Jul 1997.

[9] J. Dai, L. Lieu, and D. Rocke, "Dimension reduction for classification with gene expression microarray data," *Statistical Applications in Genetics and Molecular Biology*, vol. 5, no. 1, Feb 2006.

[10] 3GPP, "Universal Mobile Telecommunications System (UMTS); LTE; Universal Terrestrial Radio Access (UTRA) and Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Measurement Collection for Minimization of Drive Tests (MDT); Overall description; Stage 2," TS 37.320.

[11] E. J. Khatib, A. Gómez-Andrades, I. Serrano, and R. Barco, "Modelling LTE solved troubleshooting cases," *Journal of Network and Systems Management*, Online, 2017.

[12] S. Fortes, R. Barco, A. Aguilar-García, and P. Muñoz, "Contextualized indicators for online failure diagnosis in cellular networks," *Computer Networks*, vol. 82, no. Supplement C, pp. 96–113, 2015.

[13] X. He, D. Cai, and P. Niyogi, "Laplacian score for feature selection," in *Proceedings of the 18th International Conference on Neural Information Processing Systems*, ser. NIPS'05. Cambridge, MA, USA: MIT Press, 2005, pp. 507–514.

[14] T. Rückstieß, C. Osendorfer, and P. van der Smagt, *Sequential Feature Selection for Classification*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 132–141.

[15] W. Yang, K. Wang, and W. Zuo, "Neighborhood component feature selection for high-dimensional data," *Journal of Computers*, vol. 7, no. 1, pp. 161–168, 2012.