Automatic Feature Selection Technique for Next Generation Self-Organizing Networks

Author 1, Author 2, Author 3, Author 4 and Author 5

***Abstract*—Despite self-organizing networks (SONs) pursue the automation of management tasks in current cellular networks, the selection of the most useful performance indicators (PIs), used as inputs for SON functions, is still performed by network experts. In this letter, a novel supervised technique for the automatic selection of PIs for self-healing functions is proposed, relying on the dissimilarity of their statistical behavior under different network states. Results using data from a live network show that the proposed method outperforms an expert's selection, allowing the volume and complexity of both network databases and SON functions to be reduced without an expert's intervention.**

***Index Terms*—Self-organizing networks, dimensionality reduction, feature selection, performance indicator, fault diagnosis, feature engineering, root cause analysis.**

I. Introduction

THE goal of self-organizing networks (SONs) is the automation of cellular networks management, supporting network operators in their daily tasks in an attempt to lower the failure-to-response time, as well as reduce both the operational and capital expenditure (OPEX and CAPEX, respectively) [1]. To that end, cellular networks are steadily monitored by means of performance indicators (PIs). That is, a set of features allowing self-optimization and self-healing tasks to be performed.

As the complexity of the network increases, the number and variety of PIs grow as well. 5G networks are expected to include several wireless technologies, novel functionalities and different and more specific service categories. Therefore, it is expected that in a short time period a huge amount of sources of information in the form of PIs will be available in order to manage an increasingly complex system. However, this fact poses several problems. First, it implies a storage issue, as the size of the databases of the OSS (operations support system) might increase consequently, leading to an increase in the CAPEX. Second, the methods implementing SON functions may experiment a performance degradation, due to the sparsity of the high-dimensional data that they have to manage. Thus, an additional step is needed: the reduction of the dimensionality of these data with the least possible loss of information.

A common approach is the identification of the most relevant PIs (called key performance indicators, or KPIs), being the most sensitive PIs to a certain change in network and thus, allowing identifying it with the highest degree of confidence. Currently, this is manually done by network experts (e.g., in [2]). However, this poses two main shortcomings. First, this is a highly time-consuming task, as it requires them to analyze a constantly growing number of network PIs, evaluating the possible relationships among them. And second, the resulting selection tends to be biased: network experts normally use a set of KPIs that, to their knowledge, best represents the network state, but this set often differs from that providing an optimal behavior for SON functions.

Pushed by the progressive deployment of novel computation techniques in OAM (operation, administration and management) tasks, some steps have been taken in the recent years towards the full automation of SON through KPI selection. In [3], a supervised correlation-based technique for KPI selection is used in order to help identifying different traffic patterns in a UMTS (universal mobile telecommunications system) network. In [4], a supervised technique based on a genetic algorithm is proposed for KPI selection in a problem of automatic diagnosis. However, the KPI selection is dependent on the algorithm used for the subsequent diagnosis. In [5], an unsupervised technique for KPI selection is proposed. In practice, the usage of unsupervised or supervised techniques depends both the availability of labeled samples and on the network experts' criteria. Provided that labeled datasets are available, different paths may be followed. If network experts simply want the error rates to be minimized, supervised techniques for feature selection are usually preferred. Nevertheless, if they want to find features revealing underlying network states that were not initially considered among the labels, unsupervised techniques are used. This is especially relevant for networks still under deployment, devoting the supervised techniques to stable and mature networks.

In this paper, a supervised method for automatic KPI selection for self-healing tasks is proposed. In particular, the main contribution is the proposal of a *filter-type* method for KPI selection applied to root cause analysis (RCA). *Filter-type* methods for feature selection are particularly efficient, being less prone to over-fitting than *wrapper-type* techniques (e.g., [4]) and also providing a KPI selection suitable for any kind of subsequent classification or prediction technique. The proposed method relies on the assessment of the statistical dissimilarity of a given PI conditioned to different underlying network states.

II. Problem formulation

OSS databases are in charge of storing network performance information. This information may be seen as a set of $N+1$-dimensional samples, $x = \{x_1, x_2, \ldots, x_N, c\}$, being N the number of monitored PIs, plus a possible additional label, c, corresponding to the network state under which each sample was gathered. Many SON functions (particularly, self-healing

The authors are with Anonymous University.

This work has been partially funded by Funding Institution 1.

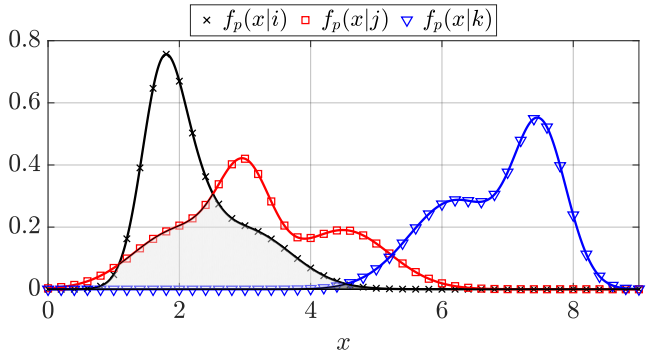

Fig. 1. PDFs of PI p conditioned to network states i, j and k, together with the overlap regions of $f_p(x|i)$ with both $f_p(x|j)$ and $f_p(x|k)$.

functions), make use of both the performance information and this label in order to detect and diagnose possible fault causes, taking the necessary actions afterwards. However, usual values for N (several thousands) pose both a storage and performance issue for the OSS databases and subsequent SON functions, respectively. In this way, a tool for dimensionality reduction appears as a pressing need in OAM tasks.

In this context, automatic detection and diagnosis (also called RCA) tasks may be formulated as a classification problem: PIs are the features of the samples, and the network state label is the class label, c. In a classification problem, a feature selection technique is a procedure that finds a subset of q features (with $q << N$) that represents a sample with the minimum loss of useful information. That is, after feature selection has taken place, the classifier takes $\tilde{x} \in \mathbb{R}^q$ as its input, instead of $x \in \mathbb{R}^N$.

III. Proposed Method for Automatic Selection of KPIs

The proposed method for KPI selection relies on how differently PIs behave in the presence of different network states. This concept can be quantized for PI p and network states i and j as the overlapping area of the PDFs (probability density function) of p conditioned to i and j: $f_p(x|i)$ and $f_p(x|j)$, respectively. This overlapping area can be mathematically expressed as Eq. (1) (from [6]), where $OVL(i,j,p)$ stands for the overlapping area for PI p when the network states i and j are considered. As an example, Figure 1 shows $OVL(i,j,p)$ (in light grey) and $OVL(i,k,p)$, which is almost negligible. In light of this, PI p would be useful to discern between network states i and k, given its different behavior under both network states, but a bad choice in order to differentiate between i and j, given its similarity.

$$OVL(i,j,p) = \int \min(f_p(x|i), f_p(x|j))dx \qquad (1)$$

Algorithm 1 describes the proposed method for KPI selection to discern among a set of network states. This procedure computes the overlapping area for each PI for each pair (i,j) of network states with $i \neq j$. After $OVL(i,j,p)$ is computed for every p, a filter is applied: only those indicators with an overlap area lower or equal to a user-defined threshold

Algorithm 1 Automatic selection of KPIs

```
for i ∈ list of states do
    for j < i do
        for p ∈ list of PIs and ∉ list of selected KPIs do
            Compute OVL(i, j, p) (Eq. (1))
        end for
        Keep only those PIs with OVL(i, j, p) ≤ T
        Sort OVL(i, j, p) in ascending order
        n first PIs → list of selected KPIs
    end for
end for
```

(T) are retained, in order to discard all the indicators that behave in a similar way under the network states i and j. Then, the resulting indicators are sorted by $OVL(i,j,p)$ in an ascending way, taking the n first. This way, the original set of performance indicators (of size N) may be reduced to a set of up to $n \cdot C(I,2)$ selected indicators, where I stands for the total number of network states and $C(\cdot)$ stands for a binomial coefficient. As a result, the computational complexity of the proposed method is proportional to I^2N.

Now, for Eq. (1) to be computed, the PDFs for each PI and network state must be estimated first from a set of M_f samples. The PDF estimate of performance indicator p under the presence of the network state i is $\hat{f}_p(x|i)$, contrary to the true PDF of such indicator, being $f_p(x|i)$. In this paper, in order to make the proposed method as autonomous as possible, a non-parametric technique for PDF-estimation is used: the kernel density estimation (KDE). In KDE, the PDF estimate is computed as the weighted sum of kernel smoothers:

$$\hat{f}(x) = \frac{1}{M_f \cdot h} \sum_{m=1}^{M_f} K\left(\frac{x - X_m}{h}\right) \qquad (2)$$

Kernel smoothers are non-negative real-valued integrable functions and are usually expressed as $K\left(\frac{x-X_m}{h}\right)$, where x stands for the random variable whose PDF is to be estimated; X_m is the actual value of the m^{th} sample of the random variable x, and h is a smoothing factor, often referred to as the bandwidth. High values for h result in high bias and low variance for the estimate $\hat{f}(x)$ and vice versa. Besides, as h decreases, kernel smoothers become narrower around the samples. If h was sufficiently small compared to the spacing among the samples, then the overlap would tend to zero. Its impact in both the accuracy of the estimation of $\hat{f}(x)$ and the trade-off between bias and variance has led to a wide variety of bandwidth selection techniques. In this paper, the value for h for kernel smoother s, h_s, is computed and optimized (leading to h_s^*) through a leave-one-out cross-validation (LOOCV) procedure, prior to the PDF fitting. h_s^* is chosen from a range of values for h_s, following the maximum likelihood (ML) criterion (Eq. (3)). The likelihood for a given value of h_s, $L(h_s)$, is computed with Eq. (4), using the LOOCV. In this equation, u stands for the index of the left-out sample; v, for the index of the remaining samples, and M_{cv}, for the number of samples devoted to the cross-validation procedure. Note that

M_f and M_{cv} sum up to the total number of samples used by the selection technique.

$$h_s^* = \arg\max_{h_s} [L(h_s)] \tag{3}$$

$$L(h_s) = \frac{1}{M_{cv}} \sum_{u=1}^{M_{cv}} \log \left[\frac{1}{(M_{cv}-1)h_s} \sum_{v \neq u} K_s \left(\frac{X_u - X_v}{h_s} \right) \right] \tag{4}$$

IV. Performance Analysis

A. Experiment setup

To evaluate the performance of the proposed method in the field of self-healing tasks within SONs, a test has been carried out using a 359-sample dataset, gathered from a live cellular network [7]. Given the lack of 5G commercial deployments at the moment of writing, an LTE (Long-Term Evolution) RAN (radio access network) has been assessed instead, without loss of generality. Each sample is composed of 286 RAN PIs and a *ground truth* label, accounting for the network state under which the sample was collected. In particular, four different labels are differentiated: high traffic (referred to as C_1), no traffic (C_2), high CPU utilization (C_3) and low coverage (C_4). The PIs in this dataset range from mobility-related counters, to retainability- and throughput-related indicators.

In this test, the performance of different feature selection techniques are compared by means of evaluating the diagnosis error rate (DER) resulting from using a diagnosis technique which takes as its input only the KPIs chosen by such selection techniques. The DER is defined as the ratio of misclassified samples to the total number of samples. In this case, a linear discriminant analysis (LDA) classifier has been used as the diagnosis method.

Seven different situations for feature selection are distinguished. First, to set a baseline, all the PIs are used, representing the situation when no selection is performed. Second, two troubleshooting experts (abbreviated hereafter as TE_1 and TE_2) are asked to select those KPIs that, to their knowledge, best represent the set of network states contained in this dataset. Next, the technique for feature selection proposed in [5] (abbreviated as UP onwards) is used, as an example of unsupervised techniques for feature selection. Then, two well known supervised techniques for feature selection are used: the ReliefF algorithm [8] (abbreviated as RL onwards) and a sequential feature selection technique [9] (abbreviated hereafter as SQ). Finally, the proposed technique (abbreviated hereafter as OV, for overlap index) is assessed.

The dataset has been split into three subsets, following a 0.4, 0.4, 0.2 proportion. The first, the *selection set*, is devised for the KPI selection; the second, the *training set*, is used to train the diagnosis method, and the third, the *test set* is used for the computation of the DERs. Note that only the indicators selected after using the first subset have been used in the training and test subsets. Given the small amount of samples in the original dataset, a Monte Carlo cross-validation (MCCV) with 50 iterations has been performed, shuffling the samples assigned to each subset in each iteration. After each split, a standard score normalization is applied over each subset.

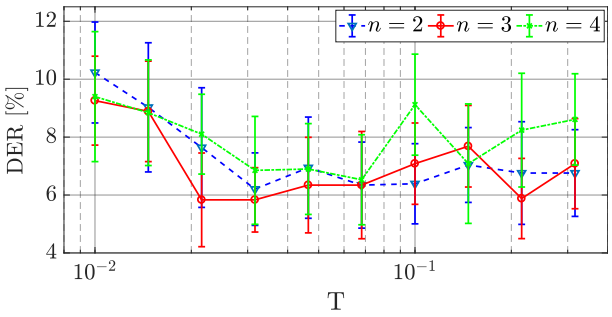

Fig. 2. DER versus T for different values of n using the Epanechnikov kernel, as a preliminary test to fine tune the parameters of the method.

For OV, the Epanechnikov kernel smoother has been used. Beside this, before comparing its performance with those of the other selection methods, a preliminary test has been performed to adjust T and n for a minimum DER. Both in this preliminary parameter tuning test and in the final test, the *selection subset* has been further subdivided into two equally sized subsets (having $M_{cv} = M_f$): one for the computation of h_{Ep}^* using Eqs. (3) and (4), and one for the PDFs fitting and KPI selection, following Algorithm 1. In this test, T has been ranged from 0.01 to 0.3, and n, from 2 to 4. For each sampling point, a 30-iteration MCCV has been performed on the original dataset, following the same 0.4, 0.4, 0.2 split. The resulting DER versus T and n is shown in Figure 2 with an error bar plot. On the one hand, low values for T lead to an increment in the DER, as it is quite likely that only one PI is chosen to discern between a given pair of network states. On the other hand, high values for T may degrade the DER (specially for $n = 4$), given the increasing probability of some low-relevance indicator to be included for a certain pair of network states. For the final test, n has been set to 2 and $T = 0.03$, due to the stability of the DER with T and the high data volume reduction achieved in that case: at most, up to 12 KPIs may be selected, out of a total of 286. To make a fair comparison in the final test, the number of KPIs being selected by UP, RL and SQ has also been limited to 12. Finally, Table I shows the number and variety of KPIs chosen by TE_1 and TE_2.

B. Results and discussion

Figure 3 shows the resulting DERs of this test over the 50 repetitions by means of a box plot for each case of KPI selection. In each box, the horizontal lines represent quartiles one (Q_1) to three (Q_3). The outer horizontal lines stand for the lower and upper adjacent values. Outliers are shown as crosses, and are identified as those samples beyond $Q_1 - 1.5 \cdot IQR$ or $Q_3 + 1.5 \cdot IQR$, respectively, where IQR is the interquartile range. For *All*, the whole feature set has been used (286 PIs). In light of these results, TE_1 managed to get a similar DER to that of *All* by only using 12 KPIs, meaning that a manual troubleshooting can be made without noticeably impacting the diagnosis performance. However, TE_2 only selected 6 KPIs. In this case, the use of so few KPIs leads to a degradation of the DER when compared to both TE_1 and *All*. It can be seen how the lack of relevant information has a bigger impact on the

DER than the degradation due to the noise-like contribution of many of the indicators in *All*. Next, it is shown how UP yields a similar performance to that of supervised techniques like RL and SQ, being all of them better than both TE_1 and TE_2, due to the avoidance of human bias.

Finally, it can be seen how OV clearly outperforms the remaining cases in terms of the median DER. In this way, when compared with the case *All*, the proposed method is shown to be capable of reducing the dataset size at least a 96%, while lowering the DER a 50%. In particular, the misclassification rates are shown on a network state basis for the cases *All* (Table IIa) and OV (Table IIb) with two normalized confusion matrices. In these tables, C_i stands for the actual *ground truth* label, whereas C_i' stands for the predicted network state. According to these matrices, only the *no traffic* (C_2) and *high CPU utilization* (C_3) cases have been almost perfectly identified in both situations. Regarding Table IIa, the case *All* shows that C_1 and C_4 (*high traffic* and *low coverage*, respectively) are often confused. This may be due to the assessment of PIs whose statistical behavior does not differ enough from one network state to another. For example, C_1 may be confused with C_4 when counters like the *number of RRC (radio resource control) connection attempts* are assessed, being relatively high in both cases. Table IIb shows how this confusion has been fully removed for (C_4, C_1') and noticeably reduced for (C_1, C_4'), highlighting the benefits of the proposed method. Besides, Table IIa shows another source of confusion, the element (C_4, C_2'): a prediction of *no traffic* given a problem of *low coverage*, due to the assessment of PIs like the *number of successful RRC connections*. This PI presents low values for both C_2 and C_4. In this case, Table IIb shows how this confusion still persists after the KPI selection. The reason for the proposed method to have chosen such PI is that, despite it behaves in a similar way according to C_4 and C_2, it behaves differently according to the pairs (C_1, C_2) and (C_1, C_4), being a good option to discern between these network states. This issue could be addressed using more stringent criteria in Algorithm 1. For example, by bounding the admissible overlap for a given PI for every pair of network states. However, this would noticeably reduce the number of KPIs (similar to the effect of a small T in Figure 2), which would lead to an eventual increase in the DER.

As a final remark, and in order to cope with the time-varying nature of the network behavior, periodical reselections of KPIs should be performed, including recently labeled samples in the sample set devoted for the KPI selection.

V. Conclusion

In this work, a method for automatic KPI selection in cellular networks for self-healing tasks has been proposed, relying on the dissimilarity of their statistical behavior under different network states. To that end, the overlap estimate for each indicator is computed for each pair of network states, using non-parametric PDF estimates. On the one hand, this technique has proven its ability to prevent network experts from the time-consuming task of selecting a set of KPIs by hand; on the other hand, it appears as a valuable tool to reduce both the storage needs and complexity of network databases and self-healing algorithms, respectively, while improving the performance of the latter.

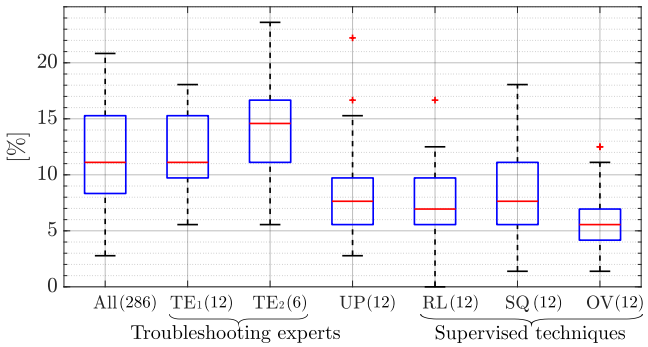

Fig. 3. DERs for an LDA classifier given different methods for the selection of the KPIs. TE = Troubleshooting Expert; UP = Unsupervised technique for feature selection [5]; RL = ReliefF [8]; SQ = Sequential feature selection [9], and OV = Overlap estimate (the proposed method). The number of KPIs being used in each selection case is shown between parentheses.

TABLE I
KPIs selected by the troubleshooting experts

1. Dropped call rate	2. #Random access attempts
3. Average channel qual. indic.	4. Retainability
5. #E-RAB[a] succ. connections	6. Handover succ. rate
7. Uplink data volume	8. Downlink traffic
9. Uplink traffic	10. Accessibility
11. #Bad cov. eval. rep.	12. CPU load 60%-80%
13. User average session time	14. Avg. no. active users

[a] E-UTRAN radio access bearer.
TE_1 used KPIs 1 to 12 from the list above.
TE_2 used KPIs 1, 3, 8, 12, 13 and 14.

References

[1] 3GPP, "Self-Organizing Networks (SON); Concepts and requirements, version 14.0.0 (2017-04)," TS 32.500.

[2] 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.

[3] J. Yang, Z. Ma, C. Dong, and G. Cheng, "An empirical investigation into CDMA network traffic classification based on feature selection," in *The 15th International Symposium on Wireless Personal Multimedia Communications*, Sept 2012, pp. 448–452.

[4] 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.

[5] 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.

[6] R. A. Stine and J. F. Heyse, "Non-parametric estimates of overlap." *Statistics in medicine*, vol. 20, no. 2, pp. 215–36, 2001.

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

[8] Z. Wang, Y. Zhang, Z. Chen, H. Yang, Y. Sun, J. Kang, Y. Yang, and X. Liang, "Application of ReliefF algorithm to selecting feature sets for classification of high resolution remote sensing image," in *2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, July 2016, pp. 755–758.

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

TABLE II
NORMALIZED CONFUSION MATRICES (SHOWN AS PERCENTAGES) AFTER DIAGNOSIS FOR THE SELECTION METHODS: (A) ALL, (B) OV

	C'_1	C'_2	C'_3	C'_4
C_1	41.7	8.3	0	50
C_2	0	94.5	0	5.5
C_3	0	0	100	0
C_4	33.3	33.3	0	33.3

(a)

	C'_1	C'_2	C'_3	C'_4
C_1	83.3	0	8.3	8.3
C_2	0	100	0	0
C_3	0	0	100	0
C_4	0	33.3	0	66.7

(b)