The SYRROCA AI-empowered network automation platform

Alessio Diamanti, José Manuel Sanchez Vilchez, Stefano Secci

The SYRROCA AI-empowered network automation platform

Alessio Diamanti*†, José Manuel Sanchez Vilchez*, Stefano Secci†
* Orange Labs, Orange, 92320 Châtillon, France. Email: firstname.lastname@orange.com
† Cnam, Paris, 75003 Paris, France. Email: firstname.lastname@cnam.fr

***Abstract*—This paper synthetically presents the SYRROCA (SYstem Radiography and ROot Cause Analysis) network automation framework at the state of the art, and details its experimental platform sufficiently enough to understand its technical demonstration. The framework aims to learn nominal operating conditions of a softwarized network service and characterize anomalies in real-time, while offering a compact system state representation called radiography. This representation can provide to operational teams with a real-time insight on anomalies at physical and virtualized layers. The related technical demonstration showcases how SYRROCA can detect real-time anomalies of different nature on a containerized vIMS (virtual IP Multimedia Subsystem) service managed by Kubernetes.**

I. Introduction

Network automation is a research area targeting the design of artificial intelligence algorithms to automate the configuration of network equipment under changing network conditions. Also called cognitive networks, such architectures aim at introducing a self-management loop able to build an abstract model of the network state evolution through data collected by specific probes. This model is then meant to be used to progressively learn the (sequence of) reconfiguration action(s) needed to cope with the events that can affect the desirable network behavior yielding to undesired state changes. The policies developed by such a cognitive/automation loop aim to adequately meet business and/or user requirements such as maintaining a certain Quality of Service (QoS), fulfilling a Service Level Agreement (SLA) [1]–[3].

Build a precise-enough model of the network state from the sensed data is a paramount step in the cognitive/automation loop. In the specific case of a softwarized infrastructure, it is composed of a large number of hardware and software components, characterizable by many heterogeneous features that can be easily extracted thanks to recent expressive monitoring tools (e.g., Prometheus). Those metrics may change in number and behavior through time and can be correlated or not to each other. The number of available and potentially valuable metrics can be extremely huge so that determining ex-post which ones are actually valuable, is not quite viable.

In this spatially complex, uncertain and varying environment, network resilience cannot be perfectly modelled. Therefore an unsupervised machine learning framework called SYRROCA (SYstem Radiography and ROot Cause Analysis) was proposed in [4] as a solution able to detect and characterize real-time anomalies in softwarized infrastructures, against a very high number of collected features. Anomalies are detected using reconstruction mean squared error (MSE) of a set of Deep AutoEncoders (DAE), each of which can detect anomalies working on real-time collected metrics regarding different type of resource. A smart representation, called radiography given its visual similarity with human radiographies, is produced to infer the impact of the detected anomalies onto the delivered network service. Similarly to known network tomography techniques [5], such radiographies give a visual view on the running and learned network behavior; however, radiographies differ in the machine learning core behind them and in that they bring insights on learned network internal characteristics using information derived from end point data. Furthermore, SYRROCA encompasses a novel root cause analysis (RCA) technique able to spot those deviating features first, and to characterize then the anomaly behaviour.

978-1-7281-7705-2/21/31.00 ©2021IEEE

II. Technical Demonstration Workflow

The demonstration showcases SYRROCA framework capabilities to: (i) learn the nominal working condition of a virtual network service using a set of unsupervised DAE; (ii) detect and characterize deviations from learned nominal states when anomalies are injected; (iii) generate real-time radiographies representations to provide a complete insight on the infrastructure state regarding different layers - physical, virtual, and service layers - and resource types - CPU, memory, storage, and network ones; (iv) characterize detected anomalies using a RCA technique able to pinpoint those features that originated the deviation from the nominal behaviour.

As depicted in Figure 1, nominal working conditions are emulated injecting a real-calls distribution across twenty days.

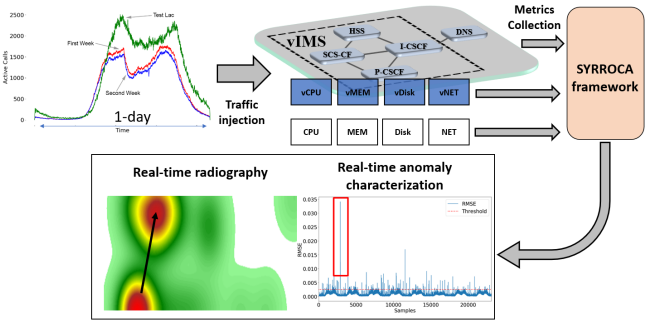


Fig. 1: Demo workflow

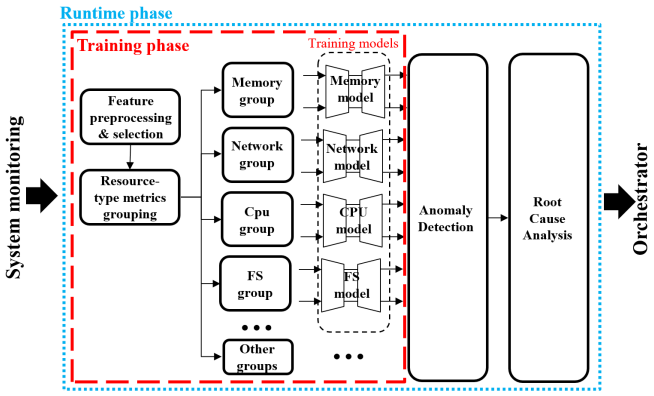


Fig. 2: High-level SYRROCA Architecture

During this phase, metrics from physical and virtual layers are collected to compose a dataset representing the nominal state. Then, SYRROCA learns through DAEs a compact representation of the dataset. In the last phase, anomalies are injected into the platform. We demonstrate how SYRROCA can detect and characterize the anomalies in real-time as long as ease RCA. To demonstrate SYRROCA capabilities we choose a virtualized IP Multimedia Subsystem (IMS) service delivering simultaneous voice calls based on real call traces. This vIMS is a containerized version managed by Kubernetes. We use the opensource OpenIMSCore IMS [6] functions, deployed as separated containers. The cluster is deployed over two physical servers equipped with an Intel(R) Xeon(R) CPU @2.10GHz with 384 GB of RAM, connected to the same network through a 1 Gbps port physical switch. The vIMS functions are deployed in a first server along with Kubernetes core components, while the second server hosts the SIPp [7] tool instances used to emulate calls among different users.

III. SYRROCA FRAMEWORK IN A NUTSHELL

The paper [4] describes the SYRROCA framework extensively, covering the machine learning algorithmic approach as well as the obtained results. The repository [8] contains the dataset used for the learning phase and the calls distributions to allow reproducibility, as well as demonstration videos.

Figure 2 depicts SYRROCA functional architecture we synthetically describe in the following with an experimental viewpoint.

A. Traffic injection and metrics collection

We emulate nominal working conditions with several SIP (Session Initiation Protocol) clients that get first registered to the vIMS core and then start a call. SIPp traffic generation tool is used to generate traffic towards the IMS to simulate calls between simulated users. Both RTP (Real-time Transport Protocol) data traffic and SIP signaling traffic are transported over UDP. We generated a real call traffic leveraging on real call logs extracted from a given LAC (Location Area Code) from Orange 3G network: we injected two weeks (March 16-29, 2020) of calls traffic following the real call distribution onto the vIMS containerized platform under test; we set the average call duration to 3 min according to [9]. Moreover, the vIMS containerized platform is tailored to correctly process this traffic load. Some examples of traffic calls distributions are shown in Fig. 1, characterized by two peaks, the first centered around 12:00 a.m and the second centered around 19:00 p.m. Degraded working conditions are simulated injecting anomalies such as packet loss, CPU overload and abnormal call distributions. For both the nominal and the degraded scenarios, we collect metrics from the physical and the virtual layers regarding the different types of resources used, i.e. CPU, memory, disk and network. We use Prometheus Node Exporter to collect metrics from the physical servers, and CAdvisor to collect metrics from the virtual Kubernetes managed layer.

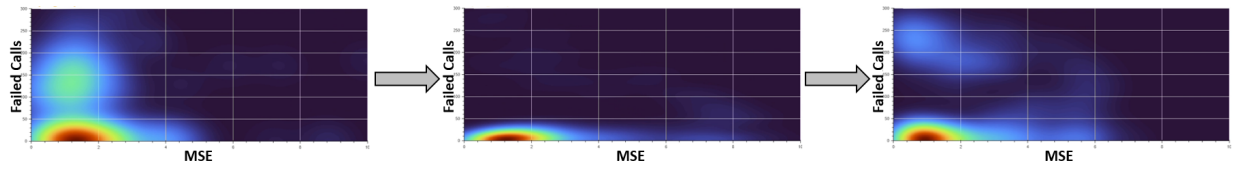
B. Learning the nominal conditions

During the learning phase, we use Deep AutoEncoders (DAE) to learn a compact and abstract representation of the simulated reference scenario. Indeed, DAEs can learn to map the input metrics to a compact representation (i.e. latent space). This is achieved optimizing its synaptic weights to minimize the reconstruction Mean Square error (MSE) when the nominal dataset is fed as input [10]. Subsequentely, when a general dataset is fed into the trained DAE, if any of the input metrics significantly deviates from the learned latent space representation, the DAE produces an higher MSE compared to the training MSE. A divide-and-conquer approach is used to group features according to the four resource type cited in the previous section. Each sub-dataset is feed to a dedicated DAE, each of them build up by Long Short Term Neural Network (LSTM NN) sharing the same architecture. Per-group dataset split makes AEs architecture design easier, reduces training time and streamlines learning. We choose LSTM NN as they are optimized to learn long term sequence correlations and to model complex multivariate sequences [11].

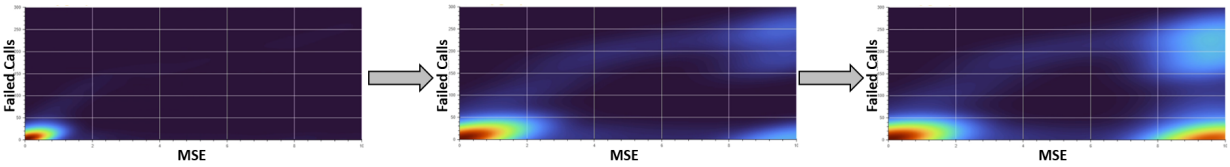
C. Real-time anomaly characterization through radiographies

SYRROCA characterizes anomalies with a deep learning unsupervised approach used to produce the compact radiography representation. The idea behind a radiography is threefold: i) to group information from all metrics in a given layer (hundreds of metrics per layer) in a compact manner; (ii) to represent the impact of deviation of those metrics on the layer above (e.g. the impact of physical layer on virtual layer and the impact of virtual layer on service layer); (iii) to track the impact of the anomalies on the system over time.

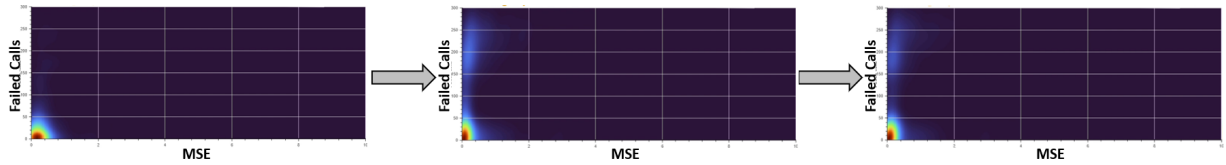
A radiography is obtained combining the MSE with a service metric to obtain a 2D density plot made with Kernel Density Estimation (KDE). In statistics, the KDE is used to estimate the probability density function of bi-variate random variables. Here we use the same technique to estimate density of the bi-variate functions $f(MSE, < service\_metric >)$ through which it is possible to locate the most frequent groups of f samples, that is the most frequent couples $(MSE, < service\_metric >)$ occurred during the considered time-window. A color scale mapping density from high to low with colors from dark to light colors, is then used to visualize the computed KDE, obtaining the so-called radiography. For



(a) Radiography evolution over time for CPU-related metrics group



(b) Radiography evolution over time for memory-related metrics group



(c) Radiography evolution over time for disk-related metrics group

Fig. 3: Radiography evolution across time

simplicity, we consider as service metric the number of failed calls, whatever is the reason causing the call failure. Therefore, it is possible to build four radiographies respectively for CPU, memory, network and disk groups. It is worth noticing that the bottom left region of the radiography corresponds to the nominal region. Whereas the further and higher density regions are located at the bottom left corner, the more significant the anomaly is. Imaging a radiography showing a dark zone for memory group corresponding to several failed call, while the remaining radiographies show dark zones corresponding to a negligible value of failed call, it is fairly intuitive to identify the root cause to be somehow linked to an anomaly impacting the memory. Figure 3 represents three sets of radiographies, each of them composed of three radiographies computed at three subsequent time-steps, from left to right, when we impose 50% of the *INVITE*'s ACK packet to be lost. The evolution in the impact of the degradation on the failed calls at each resource group is very different. We can see that the memory-related group radiography evolves towards an high density region corresponding to a degraded state located at the right bottom region, as long as a medium density region on the top right. However, this degradation is not as clearly visible on CPU-related group and nearly absent in the network related group. Indeed, the packet loss is not directly affecting network related metrics as there is no anomalous traffic, whereas lost packet generate an overhead on CPU and memory due to failed calls management.

D. Conclusion and future work

In this paper we briefly showcase how SYRROCA framework can be used to ease real-time analysis of virtualized network anomalies by radiographies and a novel RCA approach. As future work, we are enhancing the framework to provide recommendations on orchestration actions to mitigate the detected anomalies, hence to get back to the nominal region. The orchestration actions are meant to depend on the set of features/metrics that characterize the anomaly through the proposed RCA approach. Furthermore we are investigating how Variational AEs could help generalizing the latent space model extracted by the AEs.

ACKNOWLEDGEMENT

This work was partially funded by the ANR CANCAN project (ANR-18-CE25-0011).

REFERENCES

[1] David et all Clark. A knowledge plane for the internet. In *Proc. of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 3–10, 2003.

[2] Qusay Mahmoud. *Cognitive networks: towards self-aware networks*. John Wiley & Sons, 2007.

[3] Carolina Fortuna and Mihael Mohorcic. Trends in the development of communication networks: Cognitive networks. *Computer networks*, 53(9):1354–1376, 2009.

[4] Alessio Diamanti, José Manuel Sanchez Vilchez, and Stefano Secci. Lstm-based radiography for anomaly detection in softwarized infrastructures. In *ITC32*, 2020.

[5] Rui et. all Castro. Network tomography: Recent developments. *Statistical science*, pages 499–517, 2004.

[6] Openimscore. http://openimscore.sourceforge.net/.

[7] Sipp. http://sipp.sourceforge.net/.

[8] Syrroca github repository. http://github.com/SYRROCA.

[9] P. O. V. De Melo et al. Surprising patterns for the call duration distribution of mobile phone users. In *Proc. of ECML PKDD*, pages 354–369. Springer, 2010.

[10] D. Cotroneo, R. Natella, and S. Rosiello. A fault correlation approach to detect performance anomalies in virtual network function chains. In *Proc. of IEEE ISSRE*, pages 90–100, Oct 2017.

[11] P. Malhotra and et al. Long short term memory networks for anomaly detection in time series. In *Proc. of Presses universitaires de Louvain*, pages 89–94, 2015.