# SOM-based behavioral analysis for virtualized network functions

Giacomo Lanciano[*†] Antonio Ritacco[†] Tommaso Cucinotta[†]
Marco Vannucci[†] Antonino Artale[‡] Luca Basili[‡]
Enrica Sposato[‡] Joao Barata[§]

## ABSTRACT

In this paper, we propose a mechanism based on Self-Organizing Maps for analyzing the resource consumption behaviors and detecting possible anomalies in data centers for Network Function Virtualization (NFV). Our approach is based on a joint analysis of two historical data sets available through two separate monitoring systems: system-level metrics for the physical and virtual machines obtained from the monitoring infrastructure, and application-level metrics available from the individual virtualized network functions. Experimental results, obtained by processing real data from one of the NFV data centers of the Vodafone network operator, highlight some of the capabilities of our system to identify interesting points in space and time of the evolution of the monitored infrastructure.

## KEYWORDS

Self-Organizing Maps, Machine Learning, Network Function Virtualization

**ACM Reference Format:**
Giacomo Lanciano, Antonio Ritacco, Tommaso Cucinotta, Marco Vannucci, Antonino Artale, Luca Basili, Enrica Sposato, and Joao Barata. 2020. SOM-based behavioral analysis for virtualized network functions. In *The 35th ACM/SIGAPP Symposium on Applied Computing (SAC '20), March 30-April 3, 2020, Brno, Czech Republic.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3341105.3374110

## 1 INTRODUCTION

Network operators face continuously a number of challenges for evolving the offered services towards more and more complex solutions facing increasingly demanding requirements coming from the new connectivity scenarios of the future Internet.

A recent trend in network operators is the wide and massive deployment of *cloud computing* technologies, for realizing flexible infrastructures management strategies able to cope with the new challenges in the area. Indeed, IP convergence has facilitated the migration of networking services from the traditional deployment of physical appliances sized for the peak hour, towards the novel *Network Function Virtualization (NFV)* approach [8], where virtualized software functions are essentially software applications that can be deployed on a private, virtualized infrastructure of the operator.

At the heart of the management of a data center for cloud computing and/or NFV, there is an efficient distributed monitoring infrastructure that gathers continuously system-level metrics from all the physical hosts and VMs deployed in the system. This data is made available to data center automation functions and human operators, so as to enable necessary operations such as monitoring of VMs status and performance, adaptation of the available tunables to the conditions of the system and operating on the VM allocation and placement.

Among the principal concerns of data center operators we can find *anomaly detection*. The capability of detecting suspect performance degradation is fundamental to the purpose of establishing automated proactive strategies to minimize the risk of SLA violations (i.e., such that the human experts can focus their efforts in the most critical activities) or to alert the staff to start the remediation/mitigation procedures in advance. Anomaly detection is roughly defined as the problem of finding patterns that significantly differ from standard behaviors. This problem is common to a number of contexts and applications such as, for instance, fraud detection within financial transactions [6], intrusion detection in a cyber-security framework, machinery fault [10] or product quality issues detection in the industrial field [11].

Considering the wide and practical impact of the problem, many methods have been developed to face anomaly detection-related tasks. These methods use concepts taken from different disciplines like Information Theory, Statistics and Machine Learning (ML). Employed technologies and algorithms vary according to the different nature of the problem and data types to be handled. In the last years, ML techniques are gaining more and more interest for anomaly detection applications due to their robustness, flexibility and capability of learning – in a continuous manner – from data [1].

Among all the families of ML approaches suitable to anomaly detection, clustering methods have the great benefit of *not* requiring labelled data, which are not always available in practical applications. In this sub-set of approaches, Self-Organising Maps (SOM), a particular kind of neural networks, have achieved interesting results when coping with industrial data [2, 3] due to their well

[*]Scuola Normale Superiore, Pisa, Italy
[†]Scuola Superiore Sant'Anna, Pisa, Italy
[‡]Vodafone, Milan, Italy
[§]Vodafone, Lisbon, Portugal


*SAC '20, March 30-April 3, 2020, Brno, Czech Republic*

ACM ISBN 978-1-4503-6866-7/20/03.
https://doi.org/10.1145/3341105.3374110

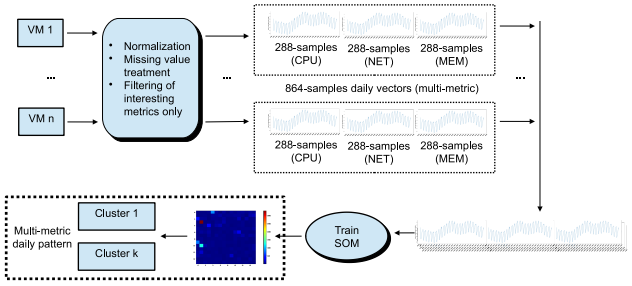


**Figure 1: Overview of the SOM-based clustering workflow.**

known capability of mapping high-dimensional data into a lower-dimensional space while preserving at clusters-level the *topology* and *distribution* of training data.

ML techniques have been successfully used in NFV applications for different purposes. For example, they have been used for an anomaly detection purposes, as shown in [4], where the performance of several supervised methods are compared by exploiting a data set containing NFV features associated to different types of faults. In [9], SOM techniques are used to detect anomalies in a small testbed in the context of NFV. General clustering techniques have also been employed in this context. In [5] the popular K-means algorithm is used to cluster cells traffic data in order to group cells with similar time behavior and allow for resource optimization. A supervised SOM-based approach is presented in [7] for fault detection, where a SOM is used to cluster *labelled* data collecting NFV performance indicators: labels are assigned by operators and used to determine whether a cluster corresponds to a faulty situation.

## 2 PROPOSED APPROACH

The goal of the proposed approach is to perform a pattern analysis of the VMs behavior, focusing on their resource consumption metrics, i.e., related to the utilization of the underlying infrastructure (*INFRA metrics*), as well as the application-level metrics (*VNF metrics*). This allows for gathering a comprehensive overview of the major behavioral patterns that characterize VMs and possibly identifying suspect behaviors. Our approach exploits the use of SOMs to cluster VM metrics patterns, leveraging on their ability to preserve the topology in the projection, meaning that similar input patterns are captured by closeby neurons (units). A VM is observed trough its movement among its best matching units (BMUs) during the time horizon under analysis, so that any changes in *"far"* BMU could be used to trigger an alarm.

The SOM-based clustering tool we realized is capable of applying clustering using multiple input metrics. In our experimentation, we have applied this technique over individual monthly data available with a 5-minutes granularity (288 observations per day, per metric, per monitored VM), amounting to several GBs of data for a specific region. The overall workflow is summarized in Figure 1. First, the raw data are preprocessed to address possible data-quality issues and to retain only the information related to the relevant metrics. The input samples to the SOM are then constructed by diving the time horizon under analysis according to a predefined period (e.g., one day) and merging the individual metrics data related to the same period in a single vector, for each VM separately. Then, such data are fed to the SOM that outputs for each of them the best matching neuron, providing a clustering.

The preprocessing phase focuses on possible issues such as (i) missing values and (ii) significant differences in the magnitude of the values of different metrics. To mitigate the effect of the first issue, a data imputation strategy is performed (e.g., linear interpolation), retaining as much data as possible for the analysis. Moreover, when using SOMs, it is recommended to address the second issue as well since, due to the sample distance evaluation mechanism, metrics with significantly larger values tend to hide the contribution of other metrics which can only take on smaller values, possibly bounded by a predefined range. To take care of this aspects, we have devised two possible strategies. The first (*normalized*) strategy consists in scaling each time-series by subtracting its mean and dividing by its standard deviation. Notice that this strategy hides information regarding the absolute magnitude of the original behavior, while it emphasizes its shape. The second (*non-normalized*) strategy consists in scaling each time-series to a range of values between 0 and 1 (inclusive) considering, for each metric individually, the historical observed minimum and maximum values. Notice that such strategy retains information regarding the absolute magnitude of the original behaviors.

Each input vector to the SOM is constituted by the concatenation of $k$ vectors, related to the preprocessed time-series of the $k$ metrics, for each considered VM and period. Given that INFRA metrics have been provided with a 5-minutes collection granularity, if a period of a day is considered, we tipically have 288 data points of each metric, for each VM, for each day. In order to train the SOM, a few *hyper-parameters* must be tuned:

**SOM dimensions**: the map is usually a finite two-dimensional region where neurons are arranged in a rectangular grid.

**Learning rate**: this parameter (between 0 and 1) controls how much each training sample contributes to the weight vector updates.

**Neighborhood radius ($\sigma$)**: this parameter (between 0 and 1) refers to the coefficient of the Gaussian neighborhood function. The higher the value of $\sigma$, the more closeby neurons are affected by the weights update of an individual neuron in each training step.

**Number of epochs**: in each epoch, neuron weights are updated in a *full-batch* fashion (i.e., the whole training data set is considered). A training process consists of a configurable number of epochs.

After the training phase, the SOM can be used to infer the BMU for each input sample, i.e., the neuron that exhibits the least quantization error when compared with the considered input sample. At this stage, the output of the analysis can be used by, e.g., a data center operator to visually inspect the behaviors captured by the trained neurons, to spot possible suspect/anomalous ones and check which VMs are associated with them. Furthermore, since the individual input samples are related to the behavior of a specific VM at specific point in time, it is also possible to analyze the evolution of the VMs throughout the time horizon, to possibly detect patterns in their behavioral changes. Additionally, we provide a mechanism to automatically detect possible suspect behaviors without the need for a human operator to inspect the status of the SOM at the end of the training. It consists of a threshold-based alert that tags an input sample as *misclassified* if it is associated to a neuron with a quantization error that is greater than a configurable threshold.

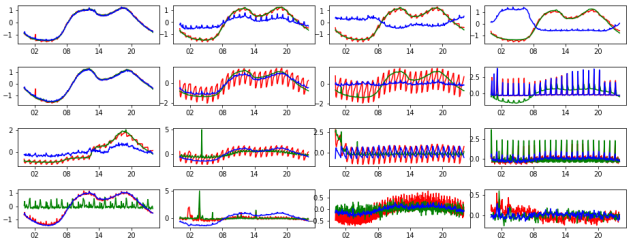

**Figure 2: Example of INFRA resource consumption clusters identified with the multi-metric SOM analysis. The red, green and blue curves in each plot correspond to the `cpu|usage_average`, `net|usage_average` and `cpu|capacity_contentionPct` metrics, respectively.**

## 3 EXPERIMENTAL RESULTS

This section provides a sample of the results that can be obtained using the approach described in Section 2. We focus over a limited set of metrics which Vodafone is currently focusing on, related to the computational, networking and storage activity of VMs and VNFs of interest. Specifically, in the following, we highlight results obtained analyzing the following metrics: `cpu|capacity_contentionPct`, `cpu|usage_average`, `net|usage_average`.

The multi-metric SOM-based analysis presented above has been applied over the INFRA metrics available for various months in one of the Italian Vodafone NFV data centers. One example output of this analysis, is the set of clusters highlighted in Figure 2. Each subplot in the picture represents the weights in the trained SOM network, that jointly identify the characterizing daily behavior for the VMs that fell into that cluster. In order to simplify presentation, the weights vectors jointly computed over the three metrics are represented overlapped but in different colors. For example, one of the most recurrent patterns is the one identified by the top-left neuron, occurring in 35.6% of the input samples.

Note that values on the Y axis can be negative because, in the preprocessing stage, we have applied a standard normalization to the input data set, as typical for behavioral analysis. This means that VMs have been clustered based on the joint shape of their daily resource consumption patterns, not their absolute values.

One of the suspect outputs we can observe in the above figure, is that the `cpu|capacity_contentionPct` figure follows closely the daily traffic pattern on the involved VMs, whereas in a healthy scenario where VMs have sufficient computational resources, we would have expected this metric to stay flat at zero, or undergo a slight increase only during the peak hours.

A significantly different pattern is the one that can be observed in the top-right neuron of Figure 2, representing 7.84% of the daily patterns observed in the month. As evident from the picture, there is a higher CPU contention during night, when the VM has lower traffic, than during the day.

Considering in input a set of identical VMs, i.e. having the same role in the VNFs and managing traffic in load sharing-mode, it was expected to obtain an identical output for all the VMs. Actually the SOM-based analysis has highlighted that a subset of such VMs is experiencing patterns very different to the standard ones that shall be monitored and further analyzed. Considering in input a set of VMs and a group of different kind of metrics (e.g. CPU, RAM and network traffic), it is possible to identify asynchronous changes among such metrics that could be linked to anomalous behavior of the NFV environment, and not only to the VNF itself.

## 4 CONCLUSIONS AND FUTURE WORK

In this paper, we presented a SOM-based technique for the classification of behavioral patterns of VM resource consumption in NFV data centers. The technique is being used across the NFV data centers of the Vodafone network operator and some of the initial preliminary results from this application have been described.

Regarding future work on the topic, we certainly have to refine the presented technique. For example, while some hyper-parameters can be effectively tuned via a grid search, others need to be tuned manually by operators, depending on the achieved results, like the misclassification threshold. An interesting possibility to consider might be the one to enrich the approach by using Deep Learning (DL) for time-series classification in order to build more effective anomaly detection models.

## REFERENCES


[1] Anna L Buczak and Erhan Guven. 2015. A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Communications Surveys & Tutorials* 18, 2 (2015), 1153–1176.

[2] L. Canetta, N. Cheikhrouhou, and R. Glardon. 2005. Applying two-stage SOM-based clustering approaches to industrial data analysis. *Production Planning & Control* 16, 8 (2005), 774–784. https://doi.org/10.1080/09537280500180949 arXiv:https://doi.org/10.1080/09537280500180949

[3] Ignacio Díaz, Manuel Domínguez, Abel A. Cuadrado, and Juan J. Fuertes. 2008. A new approach to exploratory analysis of system dynamics using SOM. Applications to industrial processes. *Expert Systems with Applications* 34, 4 (2008), 2953 – 2965. https://doi.org/10.1016/j.eswa.2007.05.031

[4] Anton Gulenko, Marcel Wallschläger, Florian Schmidt, Odej Kao, and Feng Liu. 2016. A System Architecture for Real-time Anomaly Detection in Large-scale NFV Systems. *Procedia Computer Science* 94 (2016), 491–496. https://doi.org/10.1016/j.procs.2016.08.076 The 11th International Conference on Future Networks and Communications (FNC 2016) / The 13th International Conference on Mobile Systems and Pervasive Computing (MobiSPC 2016) / Affiliated Workshops.

[5] L. Le, D. Sinh, B. P. Lin, and L. Tung. 2018. Applying Big Data, Machine Learning, and SDN/NFV to 5G Traffic Clustering, Forecasting, and Management. In *2018 4th IEEE Conference on Network Softwarization and Workshops (NetSoft)*. 168–176. https://doi.org/10.1109/NETSOFT.2018.8460129

[6] N. Malini and M. Pushpa. 2017. Analysis on credit card fraud identification techniques based on KNN and outlier detection. In *2017 Third International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB)*. 255–258. https://doi.org/10.1109/AEEICB.2017.7972424

[7] M. Miyazawa, M. Hayashi, and R. Stadler. 2015. vNMF: Distributed fault detection using clustering approach for network function virtualization. In *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*. 640–645. https://doi.org/10.1109/INM.2015.7140349

[8] NFV Industry Specif. Group. 2012. Network Functions Virtualisation. Introductory White Paper.

[9] T. Niwa, M. Miyazawa, M. Hayashi, and R. Stadler. 2015. Universal fault detection for NFV using SOM-based clustering. In *2015 17th Asia-Pacific Network Operations and Management Symposium (APNOMS)*. 315–320. https://doi.org/10.1109/APNOMS.2015.7275446

[10] R. Samrin and D. Vasumathi. 2017. Review on anomaly based network intrusion detection system. In *2017 International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECCOT)*. 141–147. https://doi.org/10.1109/ICEECCOT.2017.8284655

[11] Frenk D Van den Berg, PJJ Kok, Haibing Yang, MP Aarnts, Philip Meilland, Thomas Kebe, Mathias Stolzenberg, David Krix, Wenqian Zhu, AJ Peyton, et al. 2018. Product uniformity control-A research collaboration of european steel industries to non-destructive evaluation of microstructure and mechanical properties. In *Electromagnetic Non-Destructive Evaluation (XXI). 6 September 2017 through 8 September 2017*. 120–129.