Artificial Intelligence in virtualized networks: a journey

Silvia Fichera[1], Antonino Artale[1], Arman Derstepanians[2], Luigi Pannocchi[2] and Tommaso Cucinotta[2,*]

[1]*Vodafone Spa, via Lorenteggio, Milan, 20147, Italy*
[2]*Scuola Superiore Sant'Anna, Piazza dei Martiri, Pisa, 56100, Italy*

Abstract

This paper provides an overview of the research activities in the area of Artificial Intelligence applied to Network Function Virtualization (NFV), carried out by Vodafone, jointly with Scuola Superiore Sant'Anna of Pisa. Artificial Intelligence techniques have been used on system-level data gathered from Virtual Machines (VMs) composing a multitude of Virtualized Network Functions (VNFs), to tackle a number of problems: from traffic forecasting for capacity planning and optimization, to the off-line analysis of the daily behavior of metrics to identify possible anomalous patterns, to a Near Real time (NRT) approach for metric prediction and anomaly detection, so to trigger prompt reaction of operators of the infrastructure and services. These problems become particularly challenging in the context of the Vodafone infrastructure, spanning across several data centers for NFV throughout a dozen European Countries.

Keywords

Anomaly Detection, Network Function Virtualization, Artificial Intelligence

1. Introduction

The advent of 5G is revolutionizing the world of telecommunications, where network operators are increasingly adopting virtualization technologies and principles from the area of Cloud Computing, as key ingredients to create flexible and scalable network infrastructures. This led to the so-called Network Function Virtualization (NFV) [1], a paradigm pushing network operators to shift away from traditional physical network appliances, typically sized for peak-hour workloads, moving towards Virtualized Network functions (VNFs). These are softwarized versions of network services (i.e., packet processing for radio access, core network, security and auditing, monitoring and billing, etc.), that can be deployed flexibly and elastically on general-purpose servers, as clusters of virtual machines (VMs) providing high reliability and precise performance levels. The NFV trend in modern networking infrastructures brings also new challenges, revolving around the capability of performing accurate workload predictions, on-time anomaly detection, and optimum allocation of virtual or physical resources throughout the infrastructure [2, 3, 4, 5].

Vodafone and Scuola Sant'Anna in Pisa teamed together to tackle these challenges using novel data-based approaches from Data Science and Machine Learning. This paper is an industrial report on these research activities, highlighting some noteworthy high-level outcomes.

Precisely, it will be shown how AI/ML techniques applied to system-level metrics gathered from VMs of the Vodafone NFV infrastructure, can be used to perform workload identification and prediction. This provides the basis to identify anomalous behaviors, detect incidents in near real time for individual VNFs, and perform intelligent workload placement for capacity planning purposes. This activity has been conducted with the data collected from the European Vodafone infrastructure that involves 11 markets and hundreds of data centers.

2. Behavioral Pattern analysis

Anomaly detection is a major challenge faced by operators. It involves identifying unusual or suspicious behaviors of the system whenever it deviates significantly from normal conditions. These deviations often precede system outages, so they need to be detected on time so to promptly raise alerts or trigger automated mitigation actions. This is crucial for establishing proactive strategies to minimize the risk of violating service level agreements (SLA), letting human experts focus on critical activities.

The proposed method utilizes Self-Organizing Maps (SOM) to analyze patterns of VM metrics in data centers for NFV, to provide a visual understanding of the main behavioral patterns, promptly detect anomalies. This technique can perform a joint analysis of system-level metrics obtained from the infrastructure monitoring system (INFRA metrics) and application-level metrics obtained

Ital-IA 2023: 3rd National Conference on Artificial Intelligence, organized by CINI, May 29–31, 2023, Pisa, Italy

*Corresponding author.

silvia.fichera@vodafone.it (S. Fichera); antonino.artale@vodafone.com (A. Artale); Arman.Derstepanians@santannapisa.it (A. Derstepanians); luigi.pannocchi@santannapisa.it (L. Pannocchi); tommaso.cucinotta@santannapisa.it (T. Cucinotta)

0000-0002-8066-432X (S. Fichera); 0000-0003-2531-1721 (A. Derstepanians); 0000-0002-6250-4939 (L. Pannocchi); 0000-0002-0362-0657 (T. Cucinotta)

CEUR Workshop Proceedings (CEUR-WS.org)

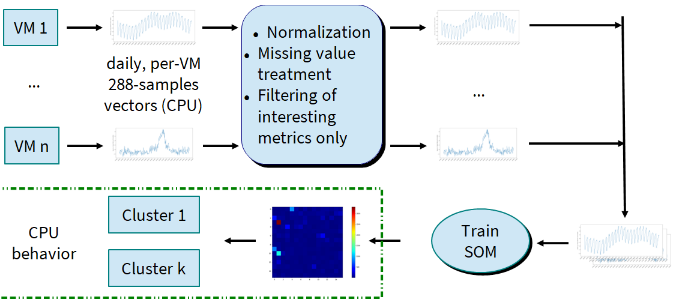

Figure 1: SOM-based clustering workflow

from individual VNFs (VNF metrics). These metrics are obtained from the NFV infrastructure manager, VMWare vRealize Operations [1], and the monitoring subsystems of the virtualized services. By jointly analyzing these metrics, we can gain a more comprehensive understanding of the major behavioral patterns of VMs and detect suspicious (anomalous) behaviors.

SOM-based clustering was performed jointly on a set of input metrics, analyzing monthly data at a 5-minutes granularity (288 samples per day per metric per monitored VM), resulting in several GBs of data per month for a specific region.

Figure 1 shows the workflow used to transform the input INFRA metrics. First, the raw data undergo preprocessing to address any data-quality issues and to retain only relevant metric information. Next, the input samples for each VM are constructed by consolidating the individual metric contributions into a single vector for each pre-defined period. The considered steps are: *i)* normalization: this was done scaling each daily time-series pattern by either subtracting its mean and dividing by its standard deviation, or normalizing to a range of values between 0 and 1 using the historical minima and maxima values observed for each metric; *ii)* missing value treatment: to address missing data and significant differences in metric magnitude, a data imputation strategy consisting of simple linear interpolation is performed; *iii)* filtering: the input data are filtered on the k specified metrics and partitioned to have a sample for each metric, VM, and period. Each input vector to the SOM is a concatenation of k vectors related to the pre-processed time-series of the k metrics for 1 day for one of the considered VMs.

Once the training phase is completed, the SOM is used to identify the best matching unit (BMU) for each input sample, providing the clustering functionality. The BMU is the neuron that has the least quantization error when compared with the input sample. This output can be used by a data center operators to visually examine the behaviors captured by the trained SOM neurons, and identify suspect or anomalous VM behaviors.

In our technique, a VM is observed through its movement among the best matching unit (BMU) during the analysis time frame. Any changes in the BMU that are distant from the previous location could indicate anomalous behavior and trigger an alarm. This enables an operator to focus on a specific set of VMs and their hosts, and conduct a further analysis that would be too time-consuming or impractical for the entire infrastructure.

Additionally, we provide a mechanism for automated detection of potential suspect behaviors. A simple threshold-based alert is triggered whenever an input sample is associated with a neuron that has a quantization error greater than the specified threshold (i.e., it is too far from its BMU). These samples are likely to represent uncommon behaviors and are marked as misclassified. The misclassified samples can be regarded as suspect or anomalous patterns that require further inspection. The misclassification mechanism can also immediately notify operators of potential misconfigurations where a too small SOM grid size has been chosen, leading to an excessive number of misclassified time series.

2.1. Grouping neurons

A noteworthy observation from using the SOM-based classification is that when employing relatively large SOM networks, the training phase often resulted in multiple adjacent SOM neurons capturing very similar behaviors. This aligns with the topology-preservation property of SOMs, which maps similar input vectors in the input space to adjacent neurons in the SOM grid. Although this can be controlled to some extent using various neighborhood radiuses, data center operators need to view a group of adjacent neurons with similar weight vectors as a single behavioral cluster. To address this, we incorporated a straightforward clustering strategy after the SOM processing stage, which combines neurons having weight vectors at a distance lower than a specified threshold into the same group. Consequently, our technique enables the consolidation of similar clusters based on the distances between the representative vectors of SOM neurons, reducing the likelihood of triggering an unnecessary alarm

[1] See: https://docs.vmware.com/en/vRealize-Operations/index.html

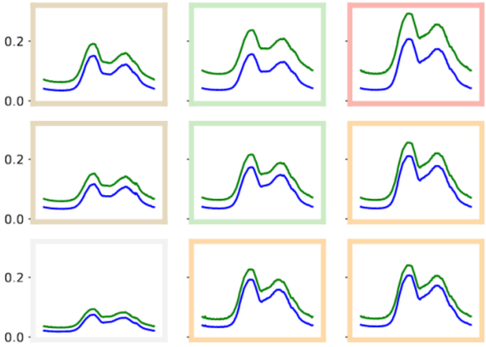

Figure 2: A 5× 5 SOM with 5 different grouped behavior

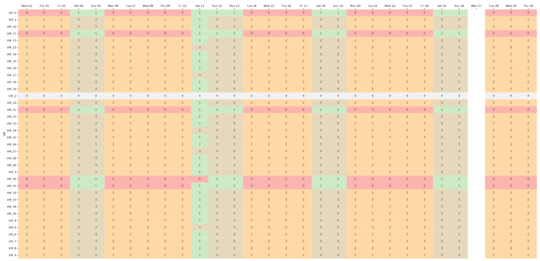

Figure 3: The behavior of each VM during April 2020

(e.g., frequent movements of a VM over time between two similar neurons) and facilitating the interpretation of results by human operators.

As an example, we have selected a data set containing CPU and NET metrics for several VMs in April 2020. Fig. 2 shows how the 5 × 5 SOM has classified the whole behaviors of different VMs in 5 groups i.e. red, orange which are regarded as high-working level groups, green and brown which are low-working and gray which is almost a flat neuron. Also, the behavior of each VM and its possible change during the whole month is shown in Fig. 3. This way, we designed an alerting system for VMs based on their daily behavior. Namely, by considering a period of time, an alert is raised once the daily behavior of a VM has been classified once in a different group. The results of such an alerting system, called "Strong Alerting System" (SAS), is shown in Fig. 4, where the dark green cells are the alerts. However, SAS showed to be prone to create a lot of false positives, since even one behavioral change raises the alert, but, as visible in the figure, several changes happen recurrently over week-ends, so they are to be regarded as non-anomalous changes. To overcome this issue, we also defined a "Weak Alerting System" (WAS), where an alert is raised when a group change occurs that is not among the weekly changes occurring every week-end. Interested readers can find more details in [6].

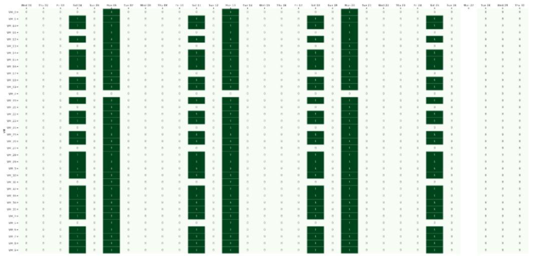

Figure 4: The Strong Alerting System for VMs on April 2020

3. Real-Time Anomaly Detection

In NFV and cloud management frameworks, anomaly detection techniques are utilized to identify issues within the infrastructure by examining the vast amount of data available through the monitoring subsystem. Real-time anomaly detection, or NRT, aims to accomplish this task promptly as soon as new data is obtained at run-time. We are dealing with metrics to be analyzed in real-time from all the NFV data centers located in 11 EU countries. The main objective is to identify anomalous points in the resource consumption and application level metrics of VMs/VNFs, which are monitored in the NFV and cloud management frameworks. Therefore, we require a scalable design, which is explained below.

3.1. System Architecture

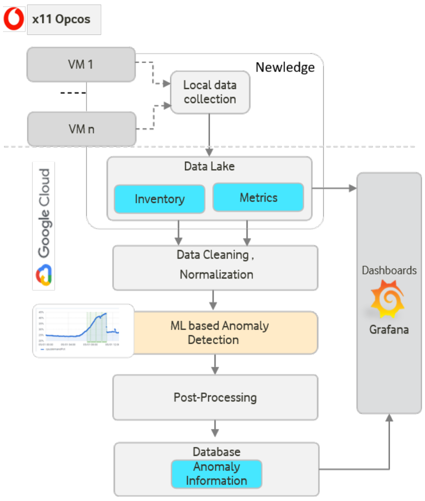

Figure 5: Anomaly detection system architecture

The NRT anomaly detection architecture we propose is depicted in Fig. 5. The data collection component gathers data from proprietary management platforms and stores it in a Data Lake within a Google Cloud Platform[2]

[2] More information at: https://cloud.google.com/.

(GCP) environment, using the Cloud Big Table[3] service as a reliable NoSQL storage for the gathered time-series. Meta-data related to all active VMs in the Vodafone virtual network infrastructure is stored in a separate SQL database, including their unique identifiers and timestamps of creation, termination or other relevant events.

The raw metrics data collected every 5 minutes for each VM are merged into a single vector for a given period and cleaned before processing with different algorithms for anomaly detection. The system is modular, allowing the configuration of various ML/AI techniques that comply with a simple interface, deployed as Google Cloud Functions[4]. These functions are triggered periodically using the Google Tasks service[5], with the output being an anomaly score for each new data point injected into the data processing pipeline since the last activation.

In the post-processing phase, single anomalous points followed by non-anomalous points are ignored, while sequences of three or more timestamps marked as anomalous are saved in a persistent storage database accessible to operators through the Grafana framework[6].

3.2. Methods and Algorithms

In order to perform NRT anomaly detection (AD), we use two main techniques: **Prediction-based AD**, where values output by a prediction model are compared with the actual samples, and, if a given threshold is exceeded, the samples are considered anomalous; **ML Algorithms** designed to directly identify anomalies/outliers, such as **Isolation Forest [7] Local Outlier Factor** [8] and **One-Class SVM** [9].

We used two main methods to perform NRT anomaly detection based on predictive models: **Long-Short-Term-Memory (LSTM)** auto-encoders, and **Simple Median (SM)**, a much simpler model based on statistical and mathematical relations among the values of the data-set, i.e., based on the ”averaged” behavior of the previous days of each VM and calculated based on a statistical median.

3.3. Results

In our context, anomalies are defined as data points whose behavior differ from the behavior exhibited previously by the same VM, as visible in Fig. 6.

We have performed comparisons among the accuracy in AD obtained with different methods and algorithms. Some AD algorithms, like Isolation Forests, were able to spot U-shape anomalous intervals like those shown in Fig. 6, only when occurring at the borders of the statistical distribution of the samples, failing to detect them in several cases, in our data sets. However, a vectorial extension of Isolation Forests was able to spot at least the beginning and ending intervals of these anomalies.

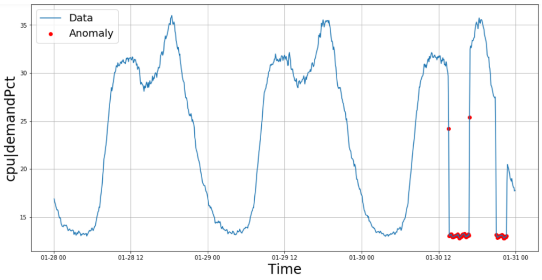

Figure 6: Spotted anomalies in NRT scenario for one VM by SM on 30th of January 2020.

For the predictive models, we identified a number of scenarios in our reference data set as particularly critical, because they were including multiple different anomalies occurring in the same day, and/or in consecutive days, sometimes spanning 3-4 consecutive days, making the analysis more challenging. In this case, the Simple Median detector we realized outperformed LSTM auto-encoders both in spotting different anomalies and producing fewer false alarms. Fig. 6 shows specifically that SM has been able to spot all anomalous points of two different anomalous intervals for a particular VM on 30th of January without any false positive detection. Details are skipped for the sake of brevity, but the interested reader can find additional details in [10], where we also made publicly available part of the data-set we used.

4. Capacity Planning for VNFs

At Vodafone, the deployment of an NFV Infrastructure consists in allocating computation workload, in the form of Virtual Machines (VMs), taking care of not exceeding the available hardware resources of the servers, considering the logistic limitations, and dealing with affinity/anti-affinity constraints on the workload. The optimal resource allocation problem has been tackled using both classical optimization, and a Genetic Algorithm.

Classical Optimization Optimization-based methods are employed when optimality guarantees are needed. Optimal placement problems are usually encoded as Mixed Integer Linear Programs (MILP) or, as Boolean Linear Programs (BLP), as done in [11]. However, whilst reliable solvers, both commercial and free, are available, MILPs and BLPs formulations suffer from the curse of computational complexity and tend to become too slow when the problem size grows. At Vodafone, we found problems that were too big to be solved optimally.

[3] More information at: https://cloud.google.com/bigtable.
[4] More information at: https://cloud.google.com/functions.
[5] See: https://cloud.google.com/tasks/docs/tutorial-gcf.
[6] More information is available at: https://grafana.com/.

Heuristics Heuristics-based methods are another common approach to the resource provisioning problem. These are usually ad hoc algorithms designed to provide a solution, following simple rules that depend on the specific problem to be solved. A lot of effort has been placed into developing Heuristics for resource allocation problems [12]. Taking advantage of the knowledge of the problem, simple heuristics reach a feasible solution faster than optimization-based approaches, clearly, at the expense of optimality.

4.1. Proposed Approach

To support Vodafone during the deployment of the Software Defined components on their network, a hybrid approach that exploits Computational Intelligence has been pursued. The success of AI techniques for solving similar placement problems is well reported in many other works, such as [13, 14, 15]. Precisely, a Genetic Algorithm has been employed to solve the resource provisioning problem, obtaining a good trade-off between solution time and the optimality of the solution. Interested readers can find more information in our prior published work on the topic [16], where the used data-set was also made publicly available.

Genetic Algorithms Genetic Algorithms are based on simple heuristics and can take advantage of the knowledge of the problem, having the possibility to avoid local minima, and thus are more likely to reach good-enough solutions. By tuning the algorithm hyper-parameters, it is possible to achieve a good performance, in terms of solution optimality, and still get a solution time comparable with the simple heuristics.

Instead of operating on a single solution, Genetic Algorithms evaluate a population of different solutions, iteratively evaluating every single solution and propagating a subset of the population (active population) selected with some criteria. A schematic representation of the algorithm is shown in Fig. 7. Often, the criteria to sort and select candidates are just how good the solution is, but there are cases where the selection also considers the population variability and some specific features. Between iterations, the selected solutions are also blended together or altered to generate new individuals that can hopefully have the good properties of the parents eventually improving with respect to them.

In the specific case of the approach used at Vodafone, the Genetic Algorithm is based on a First Fit heuristic where the processing order of the VMs to be placed is the optimization variable of the Genetic algorithm. That is, the algorithm will search for the processing order that will give the best result once placed by a First-Fit. The generation of new candidates is implemented by mutating existing solutions. The mutation randomly generates different processing orders, swapping couples of VMs, that will be tested for performance.

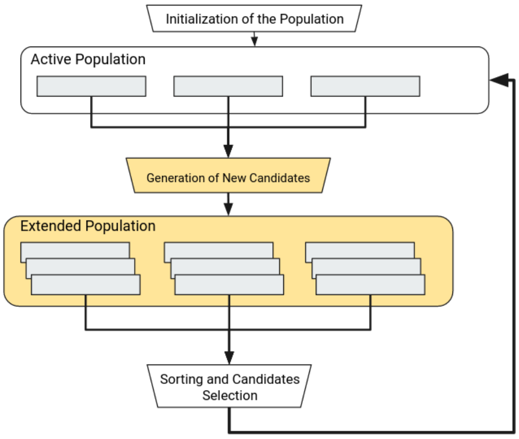
Initialization of the Population
Active Population
Generation of New Candidates
Extended Population
Sorting and Candidates Selection

Figure 7: Generic Structure of a Genetic Algorithm.

4.2. Experimental Evaluation

An experimental campaign, using real problems provided by Vodafone, has been set up to evaluate the proposed approach. The goal was to allocate the Virtual Machines to the minimum number of Hosts. The results are compared with the ones obtained from a classical BLP approach and a simple First-Fit heuristic. As expected, the proposed approach achieved a trade-off between the quality of the solution, in terms of used Hosts, and the time to get the solution. In Fig. 8 the solutions of some representative instances, for a different amount of VMs to be placed, are reported. For large problems with thousands of VMs the Heuristic approach delivers a sub-optimal solution, whilst the Genetic Algorithm's one is comparable with the optimal one returned from the MILP problem. As far as the computation time is concerned, Fig. 9 reports the time necessary to obtain the solutions reported in Fig. 8. In this case, the pattern shows that the Heuristic is the fastest approach, while the MILP approach takes the longest amount of time to return the solutions. The Genetic Algorithm stays in between. For large problems, the solution time of the Genetic Algorithm is still considered acceptable by Vodafone operators, whilst the MILP is considered too slow.

5. Conclusions

This paper provided an overview of how AI and ML techniques are being used within the Vodafone NFV infrastructure to ease and enhance a number of data-center operations related to workload prediction, anomaly detection and capacity planning.

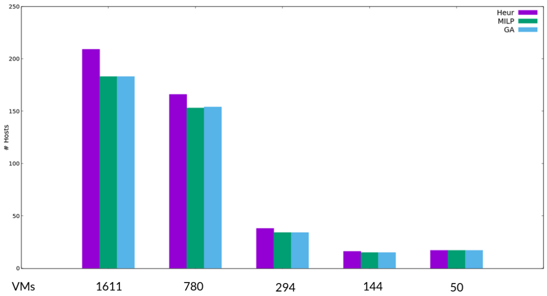

Figure 8: Number of hosts in the placement solutions found by different algorithms for problems of varaious sizes.

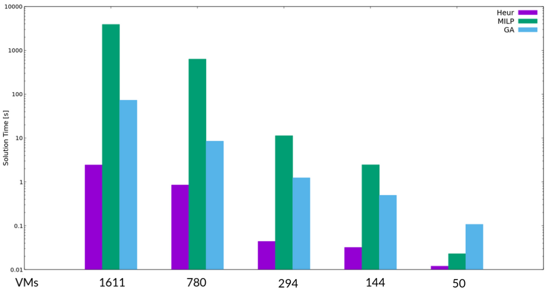

Figure 9: Comparison of the solution time of the algorithms for problems of different sizes.

References

[1] ETSI, Network Functions Virtualisation, White Paper 1, SDN and Openflow World Congress, Darmstadt, Germany, 2012. URL: https://portal.etsi.org/NFV/NFV_White_Paper.pdf.

[2] J. Gil Herrera, J. F. Botero, Resource Allocation in NFV: A Comprehensive Survey, IEEE Trans. on Netw. and Serv. Manag. 13 (2016) 518–532.

[3] F. L. Pires, B. Barán, A virtual machine placement taxonomy, in: 2015 15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, 2015, pp. 159–168.

[4] M. Xu, W. Tian, R. Buyya, A survey on load balancing algorithms for virtual machines placement in cloud computing, Concurrency and Computation: Practice and Experience 29 (2017).

[5] N. T. Hieu, M. D. Francesco, A. Y. Jääski, A virtual machine placement algorithm for balanced resource utilization in cloud data centers, in: Proceedings of the 2014 IEEE International Conference on Cloud Computing, CLOUD '14, IEEE Computer Society, USA, 2014, p. 474–481.

[6] G. Lanciano, A. Ritacco, F. Brau, T. Cucinotta, M. Vannucci, A. Artale, J. Barata, E. Sposato, Using self-organizing maps for the behavioral analysis of virtualized network functions, in: D. Ferguson, C. Pahl, M. Helfert (Eds.), Cloud Computing and Services Science, Springer International Publishing, Cham, 2021, pp. 153–177.

[7] F. T. Liu, K. M. Ting, Z.-H. Zhou, Isolation forest, in: 2008 Eighth IEEE International Conference on Data Mining, 2008, pp. 413–422.

[8] E. Schubert, A. Zimek, H.-P. Kriegel, Local outlier detection reconsidered: A generalized view on locality with applications to spatial, video, and network outlier detection, Data Min. Knowl. Discov. 28 (2014) 190–237.

[9] P. Oliveri, Class-modelling in food analytical chemistry: Development, sampling, optimisation and validation issues - a tutorial, Analytica Chimica Acta 982 (2017) 9–19.

[10] A. Derstepanians, M. Vannucci, T. Cucinotta, A. K. Sahebrao, S. Lahiri, A. Artale, S. Fichera, Near realtime anomaly detection in nfv infrastructures, in: 2022 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN), 2022, pp. 26–32.

[11] T. Cucinotta, D. Lugones, D. Cherubini, E. Jul, Data Centre Optimisation Enhanced by Software Defined Networking, in: 2014 IEEE 7th International Conference on Cloud Computing, 2014, pp. 136–143.

[12] S. Martello, P. Toth, Knapsack Problems: Algorithms and Computer Implementations, John Wiley Sons, Inc., USA, 1990.

[13] M. A. Khoshkholghi, J. Taheri, D. Bhamare, A. Kassler, Optimized service chain placement using genetic algorithm, in: 2019 IEEE Conference on Network Softwarization (NetSoft), 2019.

[14] S. Long, Z. Li, Y. Xing, S. Tian, D. Li, R. Yu, A reinforcement learning-based virtual machine placement strategy in cloud data centers, in: 2020 IEEE 22nd International Conference on High Performance Computing and Communications; IEEE 18th International Conference on Smart City; IEEE 6th International Conference on Data Science and Systems (HPCC/SmartCity/DSS), 2020, pp. 223–230.

[15] A. Jumnal, S. M. Dilip Kumar, Optimal VM Placement Approach Using Fuzzy Reinforcement Learning for Cloud Data Centers, in: 3rd International Conference on Intelligent Communication Technologies and Virtual Mobile Networks, 2021.

[16] T. Cucinotta, L. Pannocchi, F. Galli, S. Fichera, S. Lahiri, A. Artale, Optimum VM placement for NFV infrastructures, in: IEEE International Conference on Cloud Engineering (IC2E), IEEE, 2022.