Near Real-Time Anomaly Detection in NFV Infrastructures II: From SM to AGMP

Arman Derstepanians
Scuola Superiore Sant'Anna, Pisa, Italy
arman.ds1917@gmail.com

Avhad Kiran Sahebrao, Sourav Lahiri
Vodafone Intelligent Solutions, Pune, India
firstname.lastname@vodafone.com

Antonino Artale, Silvia Fichera
Vodafone, Milan, Italy
firstname.lastname@vodafone.com

Tommaso Cucinotta
Scuola Superiore Sant'Anna, Pisa, Italy
tommaso.cucinotta@santannapisa.it

***Abstract*—Our prior work on single-metric near real-time anomaly detection is extended in this paper through the generalization of a model that was initially developed for the monitoring of CPU utilization anomalies in Vodafone's Network Functions Virtualization (NFV) infrastructure. The initial generalization involves the model being adapted to other critical infrastructure KPIs, with a specific focus placed on average Network and Memory usage. Subsequently, a significant reduction in the model's free parameters is introduced, with the original count of 13 being decreased to a single parameter. Building upon this refined single-metric model, a novel multi-metric anomaly detection model is then constructed. The quality of anomaly detection is demonstrably enhanced by this model through a substantial reduction in the incidence of both false positive and false negative classifications. Empirical results from an experiment conducted on real-world data obtained from Vodafone's infrastructure are presented, with the superior performance of the newly developed multi-metric predictor being illustrated in comparison to its single-metric counterparts. The dataset utilized in this study, along with the corresponding labeled anomaly dataset, is released under an open data license to facilitate further research in this domain.**

***Index Terms*—Anomaly Detection, Network Function Virtualization, Time-Series Analysis**

I. Introduction

Cloud Computing technologies [1] are disrupting the world of Information and Communications Technologies (ICTs) in a diverse variety of application domains: web servers and online storage; social networks; on-line gaming; media management and production; on-demand video streaming; high-performance computing and GPU-accelerated Machine Learning and Artificial Intelligence computations; and others. Not only public commercial Cloud services have been taking off in the last decade, but an increasing number of ICT scenarios make use of private Cloud data centers, within which Cloud principles are reused to deploy flexible, elastically scalable, general-purpose computing infrastructures capable of supporting specific application scenarios.

Among these, it is noteworthy to mention network operators and the paradigm shift they're undergoing, with the advent of Network Function Virtualization (NFV) [2], [3] and Software-Defined Networking (SDN) [4], [5]. In this area, the high-bandwidth, low-latency and high-reliability connectivity requirements in modern and future mobility scenarios [6], [7], many of which making use of public Cloud services, caused a much higher dynamicity in networking traffic conditions than what it used to be in the past [8]. NFV and SDN are basically providing a highly flexible, software-defined, networking infrastructure that can be managed with the ease, elasticity, adaptability and automation techniques typical of Cloud environments [9]. This way, network operators are walking away from the traditional static allocation of physical dedicated network appliances, sized for peak-hour operations, to switch to a much more flexible configuration where these telecom network functions become software components, a.k.a., virtualized network functions (VNFs). These are deployed and managed as virtual machines (VMs) or containers within a set of general-purpose servers, constituting a highly flexible Network Virtual Infrastructure (NVI): here, thanks to real-time monitoring and operations, individual functions are elastically scaled to adapt in real-time to the always-changing conditions of the networking traffic to be handled.

In this context, designing smart network monitoring mechanisms is paramount [10]–[12], to realize critical network management and operations functions, including elasticity [13] and placement [14], [15] strategies, coupled with the needed real-time anomaly detection techniques [16]–[18], where metric forecasting can also play a key role [19], [20], to achieve a really "intelligent" operations engine [21].

The typical monitoring infrastructure of a Cloud or NFV data center collects and processes continuously very large amounts of data, generated in the form of several time-series per monitored instance (virtual and physical). The monitored metrics may range from system-level metrics (e.g., CPU, network, disk utilization, power consumption), to application/service-level metrics (e.g., volumes of correctly served requests, experienced error conditions, etc.), reaching easily to up to dozens of metrics per monitored instance.

One of the toughest challenges in this context, is the one to detect, in such data, anomalies that might highlight ongoing problems throughout the infrastructure, with a potential of impact on end-users, either immediately or in a short future.

a) Contributions: This paper extends a previous paper of ours [18], presenting an improved technique for near real-time (NRT) anomaly detection being used to analyze monitoring data coming from the NVI infrastructure of Vodafone, spanning across 12 EU countries (OpCos). Such data includes dozens of metrics collected with a time granularity of one sample every 5 minutes, for tens of thousand VMs deployed across thousands of physical machines. The anomaly detection system has to analyze GBs of data every month, making its design quite challenging due to the need for appropriate trade-offs between accuracy and computational overheads.

The paper describes how we extended the previous anomaly detector to deal with multiple metrics in a multi-dimensional monitoring set-up, where the different considered metrics exhibit also a great heterogeneity in their variability. It discusses key elements of the based software architecture under development, to integrate the proposed anomaly detection technique, among other AI-based tools [15], [17], [20], [22], exploiting a public Cloud provider's FaaS services to enhance scalability of the solution.

Results are presented from an experimental comparison, carried out on real data from the Vodafone NFV data centers. The presented experimental results highlight benefits and shortcomings of these techniques. Part of the dataset used for the results shown in this paper is released with an open data license (see Section VII).

b) Paper Overview: The remainder of this paper is organized as follows: Section II provides a concise overview of pertinent research in the field. Subsequently, Section III delineates the reference architecture that has been developed and deployed for NRT anomaly detection. Section IV then offers a brief review of the previously introduced algorithm employed for single-metric anomaly detection. The generalization of this algorithm to encompass all other single-metric anomaly detection scenarios, achieved through significant simplification, is presented in Section V. Following this, Section VI introduces the novel multi-metric anomaly detection algorithm, which operates based on the single-metric model detailed in the preceding section. Section VII furnishes an analysis of the accuracy and improvements in anomaly detection results through the presentation of quantitative experimental findings. Finally, Section VIII synthesizes the primary concluding remarks pertaining to the novel algorithms introduced herein.

II. Related Work

In the research literature, several authors tackled the problem of anomaly detection for NFV and cloud operations, often with the help of fault injection techniques. For example, a data set injecting faults in a Kubernetes cluster has been published in [23], where researchers evaluated different techniques for anomaly detection based on supervised machine learning (ML), including support vector machines (SVMs), nearest neighbor, naive Bayes and random forests. SVMs have also been used in [24] for on-line detection of anomalies in data from transmissions in Wireless Sensor Networks. In order to deal with transients of the time-series, here an SVM with a Gaussian kernel has been applied to data fitted with a least-squares regressor over a sliding window on the raw data to process.

An evaluation of several supervised ML techniques for off-line anomaly detection in NFV can be found in [25]. Authors compared 13 different techniques, including various types of decision trees, random forests, Bayesian networks and SVMs, on host monitoring data obtained by injecting anomalies in a test set-up running components from the ClearWater IMS system within KVM VMs deployed in an OpenStack environment. Another interesting survey can be found in [16], where authors discuss the risk of facing anomalies when switching to a NFV/cloud model, mostly due to virtualization and resource over-commitment issues causing temporal interference among co-hosted VNFs.

In the research literature on anomaly detection, works focusing on unsupervised techniques have also been proposed. For example, a technique based on Hierarchical Temporal Memory (HTM) can be found in [26] for analyzing streaming data in real-time. However, the technique was evaluated on a benchmark using single-variate data. The work in [22] proposed to use Self-Organizing Maps (SOMs) for anomaly detection in NFV data centers, with a multi-variate analysis method that identifies clusters of similar daily patterns in multiple metrics of VMs of one or more VNFs, so that changes in the classified behavior is marked as a possible anomaly. The technique was coupled with a heuristic for removing false positives, as often occurring over transitions between working and weekend days. However, the proposal focused on off-line analysis of daily behavioral patterns as observed in a recent time horizon.

A digital twin-based approach is proposed in [27] for root cause analysis of anomalies in NFV infrastructures, formulating the problem as a dynamic set-covering problem, using also hidden Markov models and transfer learning.

Compared to the techniques recalled above, in our published paper in [18] we discussed how we tackled the challenges behind detecting anomalies in near real-time, within the Vodafone NVI infrastructure. There, we presented a technique that was purposely designed around the use of a few algebraic operations, to provide a fast, lightweight and effective anomaly detection mechanism, as highlighted by the measurements we could perform on real data.

This paper presents how we extended that work, dealing with the additional issues arising from the consideration of multiple metrics in a multi-dimensional set-up, where different metrics exhibit great heterogeneity in their variability (e.g., compare the 0-100 range of CPU% with the one of the memory occupation or network bandwidth).

We hope this paper may help other industrial practitioners with the task of designing processing pipelines for anomaly detection in big infrastructures.

III. Proposed Architecture

Anomaly detection methodologies play a critical role in NFV and cloud management ecosystems by identifying irreg-

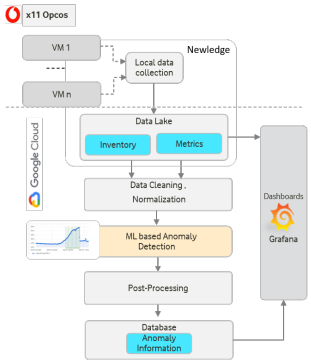


Fig. 1: Anomaly Detection System Architecture

ularities within the infrastructure, leveraging the extensive volume of telemetry data produced by their monitoring systems. Near Real-Time (NRT) anomaly detection specifically targets the rapid analysis of this data as it is ingested at run time. This proactive approach enables relevant teams to be alerted immediately to potential faults, thereby facilitating swift intervention, minimizing disruptions to the services hosted on the platform and possible economical loss for the company. In our implementation, we focus on the real-time analysis of performance metrics across NFV infrastructures deployed in data centers spanning 12 EU countries.

Our architecture, outlined in Figure 1, is tailored to identify anomalies in the temporal behavior of metrics associated with Virtual Machines (VMs) and Virtual Network Functions (VNFs). These metrics fall into two categories: infrastructure-level (INFRA) metrics that reflect the usage of underlying compute and network resources, and application-level (VNF) metrics related to the performance of the VNFs themselves. A comprehensive discussion of these metric types and their standard patterns is available in our earlier publication [22].

VM-related monitoring data is acquired from proprietary orchestration platforms through localized collection agents. This data is stored in a Data Lake hosted on Google Cloud Platform (GCP)[1], with Cloud Bigtable[2] employed as the primary NoSQL database for time-series persistence. Alongside this, we maintain a relational SQL database containing metadata about the VMs active within Vodafone's NFV infrastructure, including unique IDs and life-cycle event timestamps (e.g., creation and termination).

The collected metrics are structured as time-series with one data point every 5 minutes per VM. For analysis, data points within a selected time window are merged into vectors. A preprocessing step then ensures the data set is clean and compatible with the anomaly detection algorithms. This involves identifying and interpolating missing values—crucial for algorithms sensitive to gaps in the data—and tagging affected timestamps as potentially anomalous. Furthermore, min-max normalization is applied on a per-time-series basis to accelerate algorithmic convergence during model training or inference.

The system architecture is intentionally modular, supporting the integration of multiple machine learning (ML) and artificial intelligence (AI) models via a standardized interface. Each model processes a vector representing a specific VM and metric. These models are deployed as Google Cloud Functions, allowing us to benefit from the scalability and flexibility of the Function-as-a-Service (FaaS) paradigm for constructing serverless data pipelines. The execution of these anomaly detection routines is scheduled using Google Tasks, ensuring periodic triggering based on the arrival of new data.

The anomaly detection pipeline generates a score for each incoming data point, indicating the likelihood of it being anomalous. In the subsequent post-processing step, isolated anomalous points that are immediately followed by normal readings are discarded, under the assumption that transient issues have already been resolved. Conversely, sustained anomalies—defined as sequences of three or more consecutive anomalous timestamps—are forwarded for long-term storage in a database, making them accessible to operators. The anomalies are visualized through Grafana dashboards, enabling intuitive monitoring and diagnostics. Our system supports the simultaneous detection of anomalies across various metrics. During evaluation, we applied our methodology on VM-level metrics with a granularity of five minutes, resulting in 288 observations per day for each monitored metric and VM.

IV. SM REVIEW

We previously published [18] Simple Median (SM), a predictive model that tries to forecast the upcoming values point by point. The approach is based on the fact that, due to its repetitive nature [17] (see also Figure 2), the value of data point like p can be predicted in the most precise way based on the behavior of similar data points at the same time of the day, in the past few days. Meanwhile, careful monitoring of the behavior of the metrics reveals that, in general, we have three different classes of behaviors which are related to different days of the week. Namely, these three categories are: *I: Weekdays*, *II: Saturdays*, *III: Sundays*.

Accordingly the predicted value of the data point p will be:

$$Pred(p) = Q_2(p-24H, p-48H, p-72H, p-96H, p-120H)$$

if $p \in$ Weekdays. For the "Saturdays" and "Sundays" categories, we should go back until $p - 72H$, which will be the corresponding value of the data point considered for Saturday/Sunday three weeks ago. The main reason behind this fact, is that the amount of available data for each anomaly detection execution is the previous one month. Even if there were more extended available data if we wanted to consider the corresponding five data points for both "Sunday" and "Saturday" categories, we would have considered very old information with less relevance to our real-time anomaly detection. Figure 2 illustrates a case of a data point belonging to the category "Weekdays" on 30-01-2020.

[1]More information is available at: https://cloud.google.com/.
[2]More information is available at: https://cloud.google.com/bigtable.

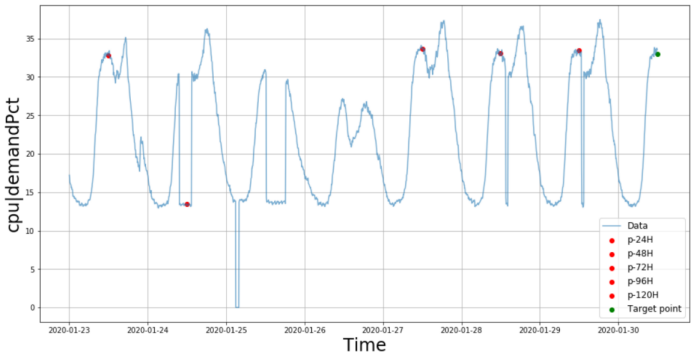


Fig. 2: Calculating the predicted value according to SM

V. Logarithmization of SM

The initial SM model was developed to spot anomalies for two key performance indicators (KPIs) related to the CPU, i.e. CPU usage average and CPU demand. After its success for this task, we decided to generalize the model to other metrics as well. In particular, our immediate targets were network usage average, memory usage average, and CPU capacity contention. However, we had the following issues:

- The SM has 13 free parameters that should be tuned based on the features of each metric separately.
- The memory usage average usually has a fluctuating behavior as its normal behavior, so we have to tune the parameters as much as possible, to skip false alarms.
- The CPU capacity contention values are very small compared to the other cases. Thus in order to have a meaningful anomaly detection according to SM we need some normalization/scaling to bring them on the same scales of memory usage average as well as CPU usage average.
- All above issues aside, the most challenging problem is the handling of Network KPI as its values are neither bounded from above nor from below, which means that they can take any positive value. Thus, no proper normalization method can work on them.

To overcome these challenges, we introduce a new version of SM that instead of taking the original values of each metric to perform anomaly detection, takes the logarithm of the values of these data points. In this case, to maintain the self-consistency of the model and avoid divergences that may occur as soon as the values are equal to 0, we add 1 to each of the values before taking the logarithm. In this way, theoretically, all values will lie in the interval $[0, +\infty)$.

Consequently, the number of free parameters is reduced to a single value, thereby simplifying the model significantly in comparison to the original SM. Furthermore, this simplification renders it a universal model applicable to all metrics. This novel model is designated the Logarithmic Simplified Simple Median (LogSSM).

VI. Multi-metric anomaly detection

Our approach for developing multi-metric predictor is based on LogSSM as a universal (single-metric) predictor for all metrics.

We perform anomaly detection on each metric individually and get the "Measure" which is the ratio of "Loss" to the "Threshold".

$$m_i = \frac{Loss_i}{Threshold_i}$$

$$\begin{cases} -1 \leq m_i \leq 1, & \text{the data point for metric i is normal} \\ \text{else}, & \text{the data point for metric i is anomaly} \end{cases}$$

where "Loss" is the difference between the "Predicted value" and "Actual value" of each metric obtained according to the execution of LogSSM for the corresponding metric.

After obtaining the "Measure" for each metric, we use geometrical methods to figure out whether a data point, from a multi-metric point of view, is anomalous or not.

Now, we use the above mentioned four values of the "Measure" columns of the following metrics:

- cpu|usage-average
- mem|usage-average
- net|usage-average
- cpu|capacity-contentionPct

to develop the multi-metric predictor. These metrics, chosen based on prior experience in behavioral pattern classification [17] and forecasting [20] at Vodafone, are described in detail in [28], [29]. Namely, we focus on a four-dimensional space where each point is represented by

$$(m_1, m_2, m_3, m_4)$$

and our objective will be to find an algebraic relation which can outline the anomalies from a multi-metric point of view. First, we divide the anomalies into two different categories.

- **Increasing**: The actual value is significantly higher than the predicted value.
- **Decreasing**: The actual value drops drastically, which in general is an indication of a severe anomaly.

Regarding the nature of cpu|usage-average and net|usage-average we expect that the detected anomalies in them should be taken more seriously than other metrics.

While among these two metrics the behavior of net|usage-average should be analyzed with the highest priority and importance.

Thus, considering the importance of m_1 and m_3 from one side and the fluctuating nature of the two other metrics i.e. "mem|usage-average" and "cpu|capacity-contentionPct" from the other side, we come up with the following equations:

$$\begin{cases} m_3 \geq 0: & (|m_1| + m_3)^2 log_{10}(max(m_2^2 + m_4^2, 10)) = \\ & 16\mathcal{H}(2 - m_1, 0)\mathcal{H}(m_1 - 1, 0)\mathcal{H}(m_3 - 1, 0) \\ -6 \leq m_3 < 0: & (|m_1| + |m_3|)^2 = \\ & 25\mathcal{H}(|m_3| - 1, 0)\mathcal{H}(|m_1| - \frac{2}{3}, 0) \\ m_3 < -6: & m_3^2 = 36 \end{cases}$$

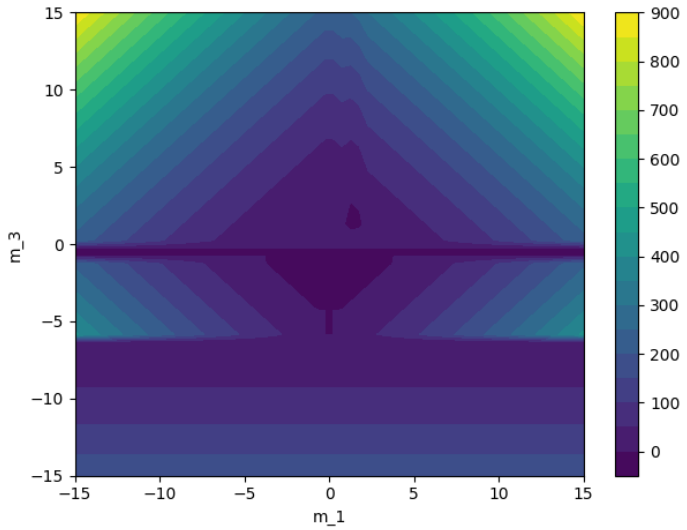


Fig. 3: Contour plot of 2D projection of AGMP

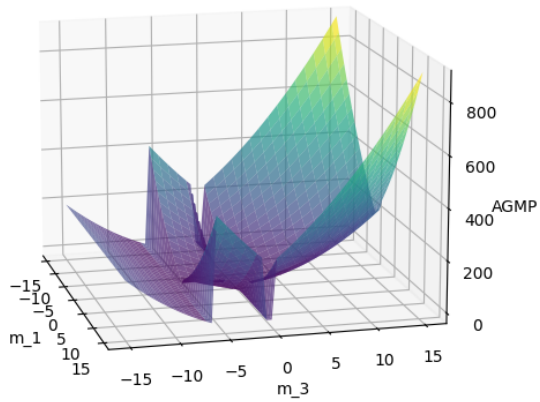


Fig. 4: 2D Projection of AGMP on the (m_1, m_3) plane

Accordingly:

If the right-hand sides of the above equations are smaller than their corresponding left-hand sides, we can conclude that it is an anomaly from a multi-metric anomaly detection point of view.

The above algebraic relation has a geometrical representation, whose two-dimensional projection has been sketched in Figures 3 and 4, and because of that is called Algebro-Geometric Multi-metric Predictor (AGMP).

The AGMP core equations stem from a comprehensive statistical analysis of historical anomaly data. We observed that severe anomalies, which typically correlate with abrupt network outages, necessitate an immediate alert; the algorithm is thus designed to prioritize these critical events. For less drastic deviations, such as significant increases or decreases from typical normal behavior, the coefficients were calibrated through a statistical examination of the "Measure" values across all four defined metrics. In these latter scenarios, particular emphasis was placed on the statistical distributions of m_3 and m_1 to refine the detection parameters.

VII. Comparison and Results

The use of AGMP allows us to detect anomalous incidents that have a severe impact on the infrastructure. This might be those cases where we have an outage on the network, or scenarios where we see sudden drops by more than one order of magnitude, happened in both CPU and network KPI in the meantime. Moreover, compared to single-metric LogSSM anomaly detection, AGMP significantly improves the results by reducing the number of unwanted False Positive (FP) detections. This is due to the specific feature of the AGMP which is called the "Asymmetry". Namely, the equations of AGMP as can be seen also in both Figures 3 and 4 have been designed in such a way to be asymmetric with respect to both m_3 and m_1 axis projections and this feature in its turn

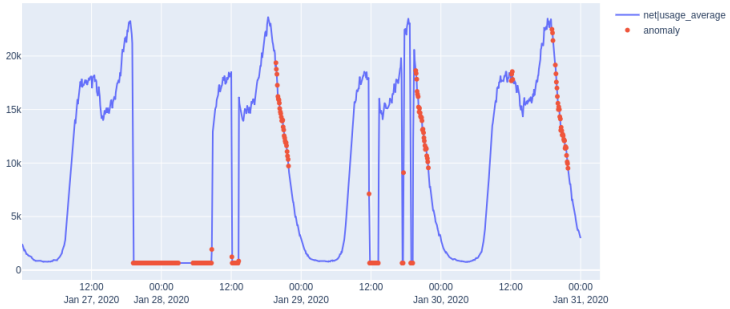


Fig. 5: Single-metric anomaly detection according to LogSSM for Network

enables us to get rid of unwanted false detections. Indeed, the problem of FP cases mostly arises when we have a recovery of the behavior after a sequence of anomalous days and intervals. This is a very common issue in real-time anomaly detection scenarios which is called "day after an anomalous day". As an example we have plotted the results of the anomaly detection a VM for its "net|usage-average" according to LogSSM in Figure 5. From this figure it becomes clear that while the LogSSM has managed to spot the anomalies successfully, it has failed to avoid FP detections which is due to the continuous presence of anomalies in the previous days. However, these false detections are not there any more once we perform a multi-metric anomaly detection. In this case as it is shown in Figure 6 the AGMP detects all anomalous points of network KPI and raises zero false alarm.

In order to test the performance accuracy of the AGMP model, we have selected a subsample of 20 VMs that contains normal and anomalous days. The dataset is released with

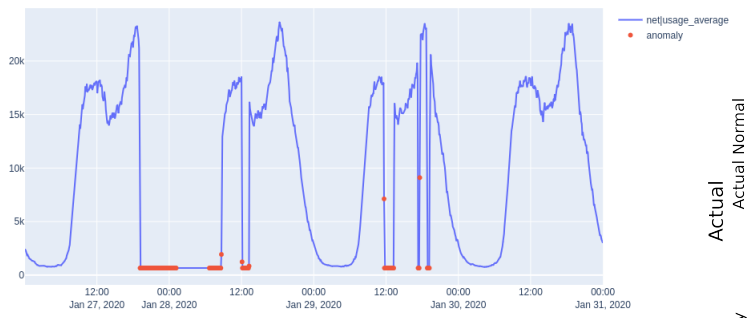


Fig. 6: Same as Figure 5 but according to AGMP

TABLE I: Scores of AGMP and two LogSSM algorithms

Predictor	Accuracy	Precision	Recall	F1 score	MCC
LogSSM - Net	0.889	0.187	0.896	0.310	0.380
LogSSM - CPU	0.951	0.335	0.753	0.464	0.482
AGMP	0.996	0.929	0.951	0.940	0.938

an open data license and is accessible at: http://retis.sssup.it/~tommaso/papers/ic2e25_ad.php. In addition , there is another dataset that includes all the detected anomalous timestamps of the selected VMs with the corresponding columns related to the values of each metric.

To outline the improvement with respect to LogSSM models, we compared the results of multi-metric anomaly detection of AGMP with two single-metric LogSSM results of the two most important metrics used in multi-metric cases i.e. the net|usage-average and the cpu|usage-average. The results of the analyses are shown in Table I. Meantime, in order to have a detailed insight about the performance of each model we have plotted their corresponding confusion matrices in the Figures 7 and 8.

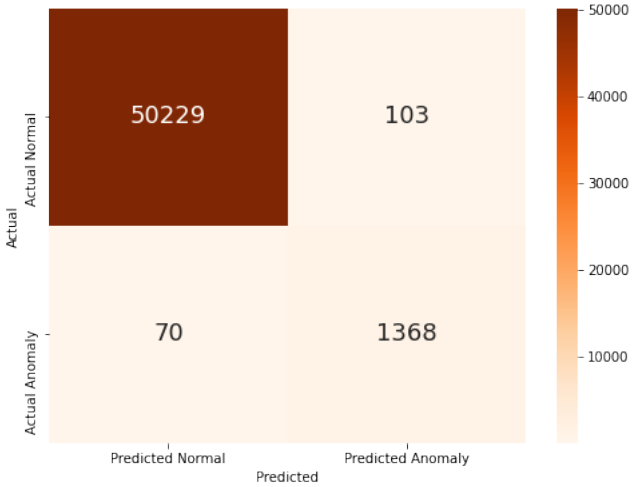
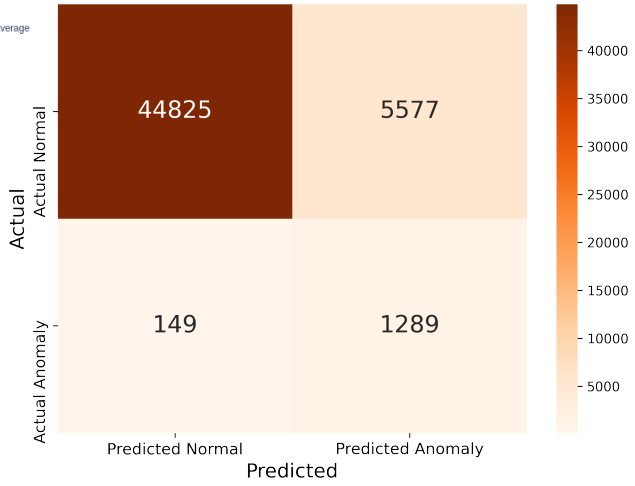
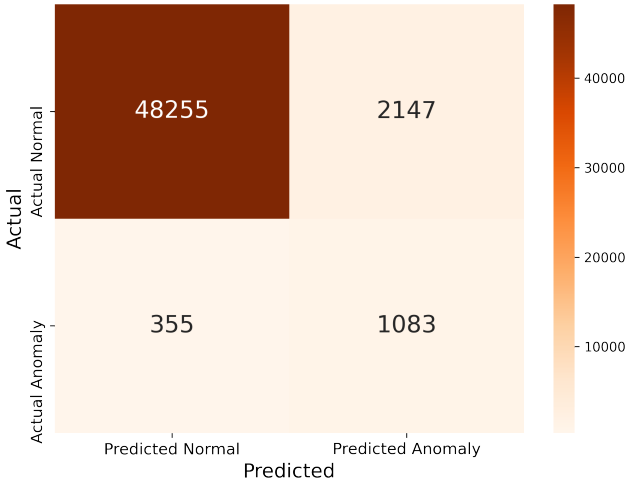


Fig. 7: Confusion matrix for the results of AGMP



(a) LogSSM on net|usage-average



(b) LogSSM on cpu|usage-average

Fig. 8: Same as Figure 7 but for the LogSSM predictor executed on network and cpu.

VIII. Conclusions

In this paper, we extended our previously obtained results in [18] of prediction based single-metric anomaly detector, i.e., the SM. We started the generalization by outlining that due to its complexity, i.e., having 13 free parameters, it is not easily applicable to other single-metric anomaly detections. In other words, for each of these detections, depending on the metric selected, we had to fine-tune the SM model. To overcome this issue, we introduced the LogSSM, which performs anomaly detection after taking the logarithm of the metric values. This not only reduces the 13 free parameters of SM to 1 but also makes the predictor a universal model applicable to all metrics

without any need for fine-tuning.

Then, we moved from single-metric anomaly detection to the multi-metric case. In contrast to classical methods where the predictor performs detection by analyzing the values of each metric, we introduced the AGMP model, which takes the "Measure" values of each single-metric LogSSM result. The key point here is that since "Measure" is the ratio of the "Loss" (the difference between the predicted and actual values) to the "Threshold," we can have a dimensionless, global value for all metrics, regardless of their standard value ranges and the presence or absence of upper or lower limits. Hence, AGMP takes the "Measure" inputs of all LogSSM detections and, based on the algebraic relation among these values, which can be represented in an abstract 4-dimensional geometrical space, determines whether a data point is anomalous or not. The introduction of AGMP as a multi-metric anomaly detection model significantly improves the quality of our detections compared to single-metric cases.

To provide readers with a better insight into the improvement of AGMP over previous LogSSM detections, we performed an experiment on a small dataset of Vodafone infrastructure. The results clearly show how multi-metric AGMP can enhance the quality of our anomaly detections compared to single-metric detections performed by LogSSM. Namely, it is shown that many FP detections of LogSSM cases are eliminated by AGMP, and many missed anomalies in single-metric detections are detected by AGMP. Finally, we should outline that, in contrast to classical ML/AI predictors, due to its purely mathematical/geometrical structure, AGMP does not require any training process, making it an ideal option for industrial projects and use-cases where reducing the time and cost of computational burden is a top priority.

REFERENCES

[1] R. Buyya, C. Vecchiola, and S. T. Selvi, "Chapter 1 - introduction," in *Mastering Cloud Computing*, R. Buyya, C. Vecchiola, and S. T. Selvi, Eds. Boston: Morgan Kaufmann, 2013, pp. 3–27.

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

[3] ——, "Network Functions Virtualisation," SDN and Openflow World Congress, Dusseldorf, Germany, White Paper 3, 2014. [Online]. Available: http://portal.etsi.org/NFV/NFV_White_Paper3.pdf

[4] Open Network Foundation (ONF), "ONF SDN Evolution," ONF, White Paper, 2016. [Online]. Available: http://www.opennetworking.org/wp-content/uploads/2013/05/TR-535_ONF_SDN_Evolution.pdf

[5] H. Woesner and D. Verbeiren, "SDN and NFV in telecommunication network migration," in *2015 Fourth European Workshop on Software Defined Networks*. IEEE, Sep. 2015.

[6] M. M. Erbati and G. Schiele, "Application- and reliability-aware service function chaining to support low-latency applications in an NFV-enabled network," in *2021 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, Nov 2021, pp. 120–123.

[7] L. Lai, G. Ara, T. Cucinotta, K. Kondepu, and L. Valcarenghi, "Ultra-low latency NFV services using DPDK," in *2021 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*. IEEE, Nov. 2021. [Online]. Available: https://doi.org/10.1109%2Fnfv-sdn53031.2021.9665131

[8] M. Gharbaoui, C. Contoli, G. Davoli, G. Cuffaro, B. Martini, F. Paganelli, W. Cerroni, P. Cappanera, and P. Castoldi, "Demonstration of Latency-Aware and Self-Adaptive Service Chaining in 5G/SDN/NFV infrastructures," in *IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, Nov 2018, pp. 1–2.

[9] P. Mell and T. Grance, "The NIST Definition of Cloud Computing," 2011-09-28 2011.

[10] G. Liu and T. Wood, "Cloud-Scale Application Performance Monitoring with SDN and NFV," in *Proceedings of the 2015 IEEE International Conference on Cloud Engineering*, ser. IC2E '15. USA: IEEE Computer Society, 2015, p. 440–445.

[11] J. Son, T. He, and R. Buyya, "CloudSimSDN-NFV: Modeling and simulation of network function virtualization and service function chaining in edge computing environments," *Software: Practice and Experience*, vol. 49, no. 12, pp. 1748–1764, 2019.

[12] R. Andreoli, J. Zhao, T. Cucinotta, and R. Buyya, "CloudSim 7G: An Integrated Toolkit for Modeling and Simulation of Future Generation Cloud Computing Environments," *Software: Practice and Experience*, Feb. 2025.

[13] G. Lanciano, F. Galli, T. Cucinotta, D. Bacciu, and A. Passarella, "Predictive auto-scaling with OpenStack monasca," in *Proceedings of the 14th IEEE/ACM International Conference on Utility and Cloud Computing*. ACM, Dec. 2021.

[14] D. Bhamare, R. Jain, M. Samaka, G. Vaszkun, and A. Erbad, "Multi-cloud Distribution of Virtual Functions and Dynamic Service Deployment: Open ADN Perspective," in *2015 IEEE International Conference on Cloud Engineering*, March 2015, pp. 299–304.

[15] T. Cucinotta, L. Pannocchi, F. Galli, S. Fichera, S. Lahiri, and A. Artale, "Optimum VM Placement for NFV Infrastructures," in *Proceedings of the 10th IEEE International Conference on Cloud Engineering (IC2E)*, Pacific Grove, California, USA, September 2022.

[16] M. Zoure, T. Ahmed, and L. Réveillére, "Network Services Anomalies in NFV: Survey, Taxonomy, and Verification Methods," *IEEE Trans. on Network and Service Management*, pp. 1–1, 2022.

[17] T. Cucinotta, G. Lanciano, A. Ritacco, M. Vannucci, A. Artale, J. Barata, E. Sposato, and L. Basili, "Behavioral Analysis for Virtualized Network Functions: A SOM-based Approach," in *Proceedings of the 10th International Conference on Cloud Computing and Services Science - CLOSER*, INSTICC. SciTePress, 2020, pp. 150–160.

[18] A. Derstepanians, M. Vannucci, T. Cucinotta, A. K. Sahebrao, S. Lahiri, A. Artale, and S. Fichera, "Near Real-Time Anomaly Detection in NFV Infrastructures," in *8th IEEE International Conference on Network Functions Virtualization and Software-Defined Networking (IEEE NFV-SDN)*, Chandler, AZ, 9 2022.

[19] Q. Sun, P. Lu, W. Lu, and Z. Zhu, "Forecast-assisted nfv service chain deployment based on affiliation-aware vnf placement," in *IEEE Global Communications Conference (GLOBECOM)*, Dec 2016, pp. 1–6.

[20] T. Cucinotta, G. Lanciano, A. Ritacco, F. Brau, F. Galli, V. Iannino, M. Vannucci, A. Artale, J. Barata, and E. Sposato, "Forecasting Operation Metrics for Virtualized Network Functions," in *2021 IEEE/ACM 21st International Symposium on Cluster, Cloud and Internet Computing (CCGrid)*, May 2021, pp. 596–605.

[21] G. Lanciano, R. Andreoli, T. Cucinotta, D. Bacciu, and A. Passarella, "A 2-phase Strategy For Intelligent Cloud Operations," *IEEE Access*, 2023. [Online]. Available: https://doi.org/10.1109%2Faccess.2023.3312218

[22] G. Lanciano, A. Ritacco, F. Brau, T. Cucinotta, M. Vannucci, A. Artale, J. Barata, and E. Sposato, "Using Self-Organizing Maps for the Behavioral Analysis of Virtualized Network Functions," in *Cloud Computing and Services Science*, D. Ferguson, C. Pahl, and M. Helfert, Eds. Cham: Springer International Publishing, 2021, pp. 153–177.

[23] Q. Du, Y. He, T. Xie, K. Yin, and J. Qiu, "An approach of collecting performance anomaly dataset for nfv infrastructure," in *Algorithms and Architectures for Parallel Processing*, J. Vaidya and J. Li, Eds. Cham: Springer International Publishing, 2018, pp. 59–71.

[24] H. Martins, L. Palma, A. Cardoso, and P. Gil, "A support vector machine based technique for online detection of outliers in transient time series," in *10th Asian Control Conference (ASCC)*, May 2015, pp. 1–6.

[25] A. Gulenko, M. Wallschläger, F. Schmidt, O. Kao, and F. Liu, "Evaluating machine learning algorithms for anomaly detection in clouds," in *IEEE International Conference on Big Data*, Dec 2016, pp. 2716–2721.

[26] S. Ahmad, A. Lavin, S. Purdy, and Z. Agha, "Unsupervised real-time anomaly detection for streaming data," *Neurocomputing*, vol. 262, pp. 134–147, 2017, online Real-Time Learning Strategies for Data Streams.

[27] W. Wang, L. Tang, C. Wang, and Q. Chen, "Real-Time Analysis of Multiple Root Causes for Anomalies assisted by Digital Twin in NFV Environment," *IEEE Trans. on Netw. and Serv. Manag.*, pp. 1–1, 2022.

[28] "vRealize Operations Manager 6.7 – vRealize Operations Definitions for Metrics, Properties, and Alerts," 2018.

[29] "VMware Aria Operations 8.18 – Virtual Machine Metrics," July 2025.