Journal of Internet Services and Applications, 2023, 14:1, doi: 10.5753/jisa.2023.3551

Anomaly Detection in Cloud-native B5G Systems using Observability and Machine Learning COTS Solutions

Gustavo Zanatta Bruno [**University of Vale do Rio do Sinos** | ***zanattabruno@edu.unisinos.br***]
Karlla B. Chaves Rodrigues [**Federal University of Goiás** | ***karllachaves@discente.ufg.br***]
Kleber Vieira Cardoso [**Federal University of Goiás** | ***kleber@ufg.br***]
Sand Luz Correa [**Federal University of Goiás** | ***sandluz@ufg.br***]
Cristiano Bonato Both [**University of Vale do Rio do Sinos** | ***cbboth@unisinos.br***]

Graduate Program in Applied Computing, University of Vale do Rio dos Sinos
Cristo Rei, São Leopoldo, RS, 93.022-750, Brazil.

Received: 14 July 2023 • **Accepted:** 29 September 2023 • **Published:** 13 December 2023

Abstract The advent of B5G networks has revolutionized the telecommunications landscape by transitioning hardware resources to software components, predominantly running on cloud-based infrastructures. However, this 'softwarization' extends across the radio access, transport, and core networks, introducing complex challenges in real-time network management. In this context of the 'softwarization', it is imperative to make the behavior of B5G systems readily observable for effective management and fault diagnosis. This article presents a comprehensive empirical investigation of observability within a B5G system, specifically focusing on its radio access and core networks. The study enhances the system's observability by combining advanced metric analysis and log parsing. Our method integrates Commercial Off-The-Shelf machine learning algorithms to diagnose anomalies and automate failure tasks. Besides that, our evaluation of the Cloud-Native Observability Tools services revealed a significant memory footprint, accounting for 86% of the total memory usage and 22% overall CPU utilization. The findings also highlight that our approach mitigates the issue of non-standardization in log data, thereby facilitating proactive failure anticipation. This study can aggregate significant value for developing automated, self-healing B5G network systems.

Keywords: Observability, 5G Systems, Metrics, Log Processing, Machine Learning, COTS

1 Introduction

Telecommunication networks are undergoing groundbreaking architectural transformations, particularly deploying fifth-generation (5G) mobile communication systems worldwide. Innovations such as edge computing, network function virtualization, network programmability, and multi-tenancy are integral to 5G. These technologies pave the way for new approaches to design, provision, and manage end-to-end services that can adapt to various needs. Additionally, 5G incorporates 'softwarization' and 'cloudification' functionalities and utilizes a fine-grained distributed system architecture, often realized through the Function-as-a-Service (FaaS) paradigm and microservices, to meet these diverse requirements.

Looking ahead, Beyond 5G (B5G) systems are set to support various applications and use cases, each with unique requirements. This diversity brings unique challenges for network observability not typically seen in traditional networks. Factors such as the large scale and increased complexity of B5G systems, device heterogeneity, dynamic network conditions, stringent security and privacy requirements, and the demand for real-time or near-real-time responses all contribute to these challenges, making existing methods from traditional networks less effective in the B5G context.

The advent of 5G networks has been a significant milestone in the evolution of telecommunication systems. As of June 2022, around 70 countries had deployed 5G networks, up from just 38 in mid-2020 Statista [2023]. Furthermore, the number of 5G users is also rapidly increasing. By the end of 2021, the total number of 5G users reached 507 million people [Gizchina, 2021]. This widespread deployment and adoption underscore the importance and complexity of managing these networks. In this context, transitioning from traditional monolithic services tightly coupled with proprietary hardware to a microservices architecture in telecommunication networks brings unique challenges. As network operators navigate this new landscape, understanding the behavior of these complex systems becomes crucial. Insufficient observability of B5G systems brings significant risks and challenges. Network operators may face difficulties detecting and diagnosing anomalies and failures without comprehensive observability, leading to prolonged downtime, degraded performance, and compromised security. However, existing literature, such as John *et al.* [2017]; Surantha and Putra [2022]; Leliopoulos and Drigas [2023], has not thoroughly explored the observability of 5G systems. Most current research focuses solely on the metric dimension of observability, neglecting other essential aspects such as logs and traces. This gap in knowledge presents a significant problem that this article aims to address, given the critical role of 5G networks in modern telecommunications.

This article addresses a research gap by offering a unique empirical analysis of how B5G system platforms make their internal states observable through standard Cloud-Native Observability Tools (CNOT). Unlike prior studies that primarily focus on individual dimensions of observability, such as metrics, or that only employ Machine Learning (ML) for anomaly detection in a 5G context, our study takes a multifaceted approach. We harmoniously combine metrics, logs, and ML techniques to view system behavior comprehensively.

We build a testbed that emulates a cloud-native B5G system consisting of the Core Network (CN) and the Radio Access Network (RAN) to validate the efficacy of these tools in metric collection, log analysis, and anomaly detection. We deploy this system as microservices on a Kubernetes (K8S) cluster using open-source platforms widely used in 5G network research, Free5GC [2023] and OpenAirInterface [2023]. Our empirical methodology distinguishes this work from the predominantly theoretical and narrow-scope practical studies in the existing literature. Furthermore, we carefully consider the structure and intelligibility of log messages generated by the system, identifying areas for improvement to make the logs more structured and easier to interpret. Most existing research in this area has focused either on specific dimensions of observability [Zheng *et al.*, 2022; Hung *et al.*, 2022] or on the application of ML techniques for anomaly detection in a 5G context [Joda *et al.*, 2022; Hakiri *et al.*, 2022], often neglecting a holistic approach that combines these elements. The main contributions of this article are:

- Designing and implementing a testbed that reproduces a cloud-native B5G system, filling a critical research gap.
- Deploying a CNOT based on standard tools to gather metrics and logs, providing a comprehensive view of the B5G system's internal states.
- Demonstrating empirically the effectiveness of a multifaceted approach that combines metrics, logs, and a Commercial Off-The-Shelf (COTS) ML solution for diagnosing anomalies in B5G systems.
- Evaluating the structure and interpretability of log messages, identifying areas that need improvement.

To the best of our knowledge, this is the first empirical contribution to shed light on the suitability of exposing the internal state of B5G systems through standard CNOTs. The rest of the article is organized as follows. Section 2 provides the necessary background for our investigation. Section 3 discusses the related work. In Section 4, we provide a detailed description of the testbed implementation. Section 5 discusses the results of our experimental evaluation. We conclude and propose future work in Section 6. Moreover, Table 2 summarizes the commonly-used abbreviations.

2 Background

This section provides a comprehensive overview of the essential principles associated with system observability, particularly emphasizing its three main components: metrics, logs, and traces. In addition, we introduce the foundational aspects of 5G systems, concentrating predominantly on the CN and RAN elements.

2.1 Observability

In this context, observability represents the capacity to perceive and understand a complex system's behavior via telemetry gathered during runtime. Distinct from traditional black-box monitoring, observability aims to decrease the time system operators take to gain a comprehensive and accurate understanding of the system's functionality and performance. Telemetry extracted from the system often falls into one of three categories: metrics, logs, and traces [Gatev, 2021]. Figure 1 illustrates these three main pillars of observability.

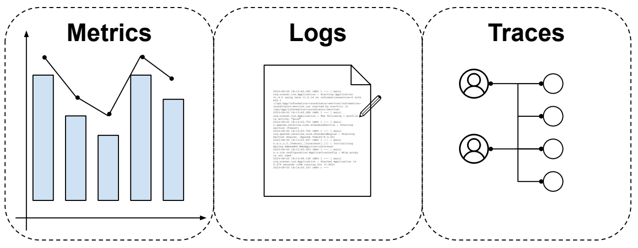

Figure 1. Three pillars of observability.

Metrics are countable or quantifiable attributes with values that are collected at consistent intervals, typically using two methods [Scrocca *et al.*, 2020]: (i) direct extraction from the operational process via a library or (ii) through the infrastructure that is executing the process. In the former case, metrics are defined and collected at the application level. In the latter case, metrics pertain to using infrastructure resources, e.g., Central Processing Unit (CPU), Random Access Memory (RAM), and communication networks. In most research, metric collection and persistence processes are considered *monitoring*, which constitutes one of the primary methods for detecting anomalies in a computing system [Gomez Blanco, 2023].

Conversely, logs are text-based records of the operations (or errors) a system performs during runtime. Such as, metrics and logs are gathered directly from operational processes. Moreover, in contrast, the generation of log records is unpredictable as they correspond with discrete events that occur at specific points in time. Logs can contain structured or unstructured information from various sources, making them rich data sources for anomaly detection and root cause analysis. However, the usefulness of logs is closely tied to the quality of the generated messages. Inefficiently produced logs can complicate fault diagnosis, require additional maintenance effort, and lead to performance degradation [Chen and Jiang, 2021].

Traces are identifiers that capture request flows within a system comprising several components, revealing the sequence of events within a system [Esteves *et al.*, 2021]. This knowledge is crucial in distributed systems where precise synchronization of nodes is often impossible. However, generating traces requires the system under observation to be instrumented. In summary, metrics, logs, and traces are the backbone of system observability, and collectively, they provide a comprehensive view of complex systems' behaviors [Li *et al.*, 2022], such as in the 5G system.

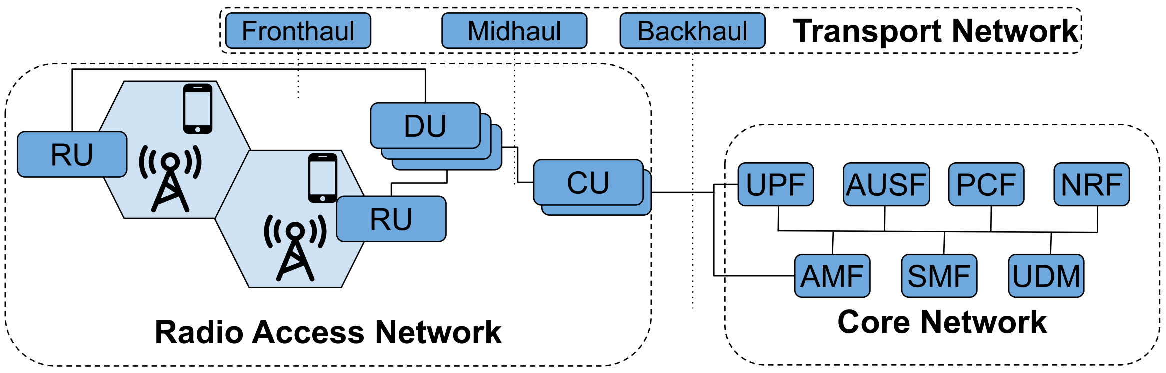

Figure 2. 5G network basic architecture.

2.2 5G System

Figure 2 depicts a 5G system consisting of RAN, CN, and the Transport Network (TN) [3GPP, 2020]. The RAN domain uses diverse access technologies to connect User Equipment (UE) and the operator's network. The Next-Generation RAN (NG-RAN), defined to meet 5G requirements, is composed of a set of Next Generation Base Stations (gNBs) providing connectivity to UE through the 5G New Radio (NR) technology [Saavedra *et al.*, 2018]. A gNB can be logically divided into three entities: Centralized Unit (CU), Distributed Unit (DU), and Radio Unit (RU) [Larsen *et al.*, 2019], with the NR protocol functions being determined by split options, leveraging the scalability and centralization benefits offered by virtualization and the FaaS paradigm [Polese *et al.*, 2023].

TN is divided into fronthaul, backhaul, and midhaul, connecting diverse components. Fronthaul is the link between the RU and the DU at the cell site, requiring high capacity and low latency for communication. Backhaul connects the cell site to CN, carrying traffic between these points through fiber, microwave, or satellite links. Midhaul, less commonly used, refers to the network segment connecting the centralized DUs and CUs. Each network segment has distinct bandwidth, latency, and reliability requirements [Klinkowski and Jaworski, 2022]. Indeed, the TN encompasses multiple technologies, including Multiprotocol Label Switching (MPLS), Dense Wavelength Division Multiplexing (DWDM), and more.

The CN allows UE to send and receive mobile traffic to and from applications hosted on the data network or the Internet. Designed with cloud-native principles in mind [3GPP, 2020; Abdulghaffar *et al.*, 2021], the 5G CN features a service-based architecture with a fully decoupled modular control plane from user plane functions. Figure 2 illustrates CN's primary network functions and a RAN disaggregated with RU, DU, and CU components. The control plane's interaction with the RAN is facilitated through the *Access and Mobility Management Function* (AMF), supporting encrypted signaling connections to UE. The *Session Management Function* (SMF) handles user session creation, modification, and termination, while user data is managed by the *Unified Data Management* (UDM) function. UE is authenticated by the *Authentication Server Function* (AUSF) using access credentials provided by UDM. *Policy Control Function* (PCF) administers policy control for access management, sessions, and mobility. Other network functions can find services through the *Network Repository Function* (NRF). Lastly, the *User Plane Function* (UPF) is responsible for forwarding and processing UE data [Cardoso *et al.*, 2020; Tang *et al.*, 2022].

3 Related Work

The literature offers varied perspectives on 5G network infrastructure, anomaly detection techniques, and ML algorithms. A standard topic in the works of [Sun *et al.*, 2020], [Doan and Zhang, 2020], [Hakiri *et al.*, 2022], [Hung *et al.*, 2022], [Yuan *et al.*, 2022], and [Kim *et al.*, 2022] is the analysis and improvement of 5G network functionality, with particular emphasis on detecting network anomalies. [Sun *et al.*, 2020] propose an adaptive rule engine for this task, while [Doan and Zhang, 2020], [Hakiri *et al.*, 2022], [Yuan *et al.*, 2022], and [Kim *et al.*, 2022] leverage deep learning and ML techniques to identify and mitigate anomalous activity.

Table 1 provides a detailed synopsis of the related work considering observability and anomaly detection, denoted by their citation keys in the ”Works” column. Moreover, the table categorizes the study into several dimensions. The ”Pillars” represent methodologies or data types employed, encompassing metrics, logs, and traces. The ”5G Systems” column indicates the practical experiments with CN or RAN components of a 5G system. The columns titled ”Container Network Function (CNF)” and ”ML” point to the implementation of these respective concepts. ”Anomaly Detection” denotes the analysis of identifying atypical patterns within network or system data. The column ”Practical Experiments” highlights whether the study incorporates empirical experiments. Moreover, an '✓' marks the characteristics of the works found in the literature.

[Sun *et al.*, 2020] present a unique 5G Mobile Edge Computing (MEC) approach by incorporating smart streetlights equipped with 5G base stations, targeting the C-RAN access network. [Doan and Zhang, 2020] use NetFlow network flow data to minimize latency impacts in the 5G CN. [Hakiri *et al.*, 2022] examines the SECRETED 5G OPS project, aiming to improve 5G network security and resilience by deploying ML algorithms and metrics and logs.

[Hung *et al.*, 2022] introduce a distributed architecture for 5G backhaul network monitoring, focusing on Wide Area

Table 1. Summary of the scope of observability and anomaly detection related work.

Works	**Pillars**			**5G Systems**		**CNF**	**ML**	**Anomaly Detection**	**Practical Experiments**
	Metrics	**Logs**	**Traces**	**CN**	**RAN**				
[Sun *et al.*, 2020]	✓			✓	✓		✓	✓	
[Doan and Zhang, 2020]	✓	✓			✓		✓	✓	✓
[Hakiri *et al.*, 2022]	✓	✓		✓	✓	✓	✓	✓	
[Hung *et al.*, 2022]	✓								✓
[Joda *et al.*, 2022]	✓				✓		✓		
[Prathiba *et al.*, 2022]	✓						✓	✓	✓
[Raja *et al.*, 2022]	✓						✓	✓	✓
[Yuan *et al.*, 2022]	✓	✓		✓				✓	
[Kim *et al.*, 2022]	✓						✓	✓	✓
[Zheng *et al.*, 2022]		✓						✓	
[Duong and Kim, 2023]	✓	✓	✓	✓		✓			
[Soldani *et al.*, 2023]	✓	✓	✓	✓	✓	✓			✓
This work	✓	✓		✓	✓	✓	✓	✓	✓

Network (WAN) monitoring. A telemetry collector with microservices architecture reduces the processing load of telemetry report data. Furthermore, [Yuan *et al.*, 2022] emphasize the importance of interpretability and reliable explanations for ML models, suggesting lightweight federate learning ML-based systems for efficient anomaly detection in the 5G CN and RAN. Similarly, [Kim *et al.*, 2022] focuses on network anomaly detection in 5G networks using domain adaptation techniques, specifically transfer learning and unsupervised domain adaptation.

The work of [Joda *et al.*, 2022] offers a distinct view by detailing the emerging Internet of Senses (IoS) field in sixth-generation (6G) networks. It focuses on semantic communications and edge intelligence to improve user equipment throughput and energy consumption. Partially Observable Markov Decision Processes (POMDP) and ML are highlighted for communication and learning metrics optimization.

[Zheng *et al.*, 2022], [Duong and Kim, 2023], and [Soldani *et al.*, 2023] discuss advanced methods for improving network performance and security. [Zheng *et al.*, 2022] focus on network traffic log analysis to detect CN and RAN anomalies. [Duong and Kim, 2023] propose a service mesh based 5G CN using various platforms, enhancing security and observability. [Soldani *et al.*, 2023] introduce an Berkeley Packet Filter (eBPF) for cloud-native observability, networking, and security in mobile networks, discussing the CN and RANs.

Finally, the works of [Prathiba *et al.*, 2022] and [Raja *et al.*, 2022] explore the role of sensor anomaly detection in autonomous vehicles in a 6G-Vehicle-to-Everything environment. [Prathiba *et al.*, 2022] introduce a hybrid deep learning framework, combining multi-agent reinforcement learning and maximum entropy inverse reinforcement learning for highly accurate anomaly detection and classification. [Raja *et al.*, 2022] propose an efficient trajectory anomaly detection and classification framework using the deep deterministic policy gradient algorithm to analyze driving patterns and detect anomalous trajectories. These two works consider various impact factors to ensure autonomous vehicles' safe and efficient functioning.

This study introduces a unique empirical examination of how B5G system platforms show their internal states using standard CNOTs, an aspect inadequately explored in the existing literature. Our study is multifaceted and diverging from prior research that primarily targets specific dimensions of observability, such as metrics, or employs ML for anomaly detection within a 5G context. It harmoniously combines metrics, logs, and ML techniques. We empirically validate the efficacy of these tools in metric collection, log analysis, and anomaly detection by establishing a testbed that emulates a cloud-native B5G system and deploying a standard cloud-native observability solution onto this platform. This empirical strategy distinguishes it from the literature's predominantly theoretical and narrow-scope practical studies. Furthermore, our study provides a meticulous assessment of the structure and interpretability of log messages, an element that needs to be addressed in the surveyed related studies.

4 Methodology and Experimental Setup

This section meticulously outlines the methodology and experimental framework employed for the empirical study. It highlights the decisions undertaken during the establishment of the testbed, encompassing hardware and software considerations. Moreover, the source code utilized in the testbed is readily available on GitHub[1] for transparency and reproducibility.

Infrastructure of the Testbed: The testbed infrastructure comprises three DELL PowerEdge M610 physical servers (pServers). Each pServer has 2 Intel Xeon X5660 processors and 192 GB RAM. These servers host a Virtual Machine (VM) running Ubuntu with a low-latency kernel. The VMs are configured with 18 vCPUs, 24 GB RAM, and a 50 GB disk. VMware ESXi 6.7 serves as the hypervisor. The servers are connected to a physical switch (pSwitch) via a virtual switch (vSwitch), emulating the TN. This network guarantees a bandwidth of 10 Gbps and a latency of 1 ms for each link. Each VM is connected to a virtual switch. A

[1]https://github.com/my5G/PMon-5G

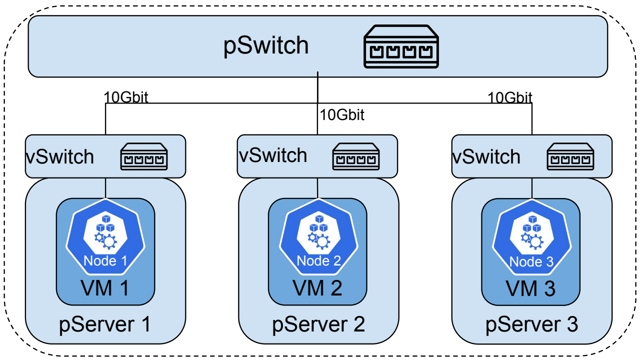

Figure 3. Testbed Infrastructure.

Kubernetes (K8s) cluster, consisting of three nodes, is deployed over this infrastructure. Each node operates within a VM, with one node functioning as the K8s master and the remaining two as K8s workers. K8s was selected for container management and orchestration due to its growing popularity among telecommunication operators [Polese *et al.*, 2023]. Calico[2], serving as the container network interface, is also employed in this setup. The testbed topology is illustrated in Figure 3.

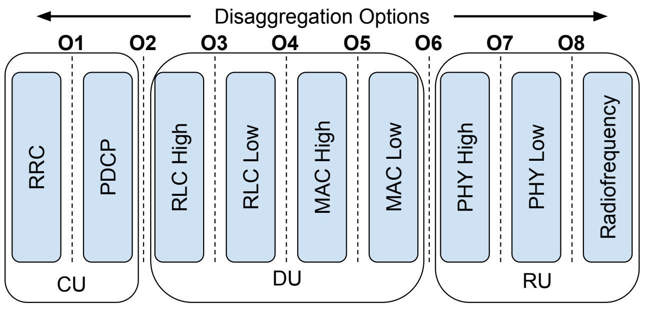

Figure 4. Disaggregation Options Used.

B5G System: OpenAirInterface[3] is utilized to emulate RAN and UE, an open-source initiative maintained by the OpenAirInterface Software Alliance that implements the 3rd Generation Partnership Project (3GPP) standard on general-purpose hardware and Software Defined Radio (SDR). Furthermore, free5GC is used to emulate CN, one of the pioneering open-source implementations of a service-based B5G core. These platforms are commonly employed in empirical studies involving B5G. Moreover, OpenAirInterface supports RAN split. Figure 4 presents the eight disaggregation options defined by 3GPP [2017]. Our experiment employs disaggregation options 2 and 6. Option 2 creates a partition in the RAN protocol stack at the Packet Data Convergence Protocol (PDCP)-Radio Link Control (RLC) boundary, effectively isolating the User Plane from the Control Plane. This architecture facilitates independent deployment and scaling of each plane. Conversely, Option 6 disaggregates the Medium Access Control (MAC) low layer from the PHY high functionalities. This specific division permits finer control over PHY-layer operations and potentially enhances system performance by allowing components to be deployed closer to the antenna. Therefore, in our testbed, RU is located in Node 3, DU in Node 2, while CU operates in Node 1, as delineated in Figure 5. The free5GC functions are installed as a single pod in Node 1. Lastly, the Physical layer is disabled while the emulated UE operates with RU to emulate RAN.

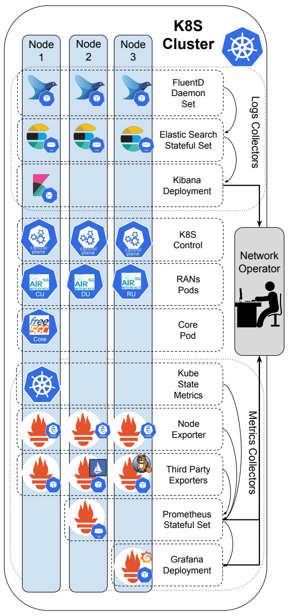

Figure 5. K8s cluster used in our empirical investigation.

Observability Software Stack: The study employs a robust observability software stack to monitor the performance and correct functioning of K8s clusters. Metrics are collected by The Prometheus[4], polling each cluster node every 30 seconds through a Node Exporter. This polling allows an accurate monitoring of cluster performance. For network-specific metrics, Prometheus utilizes third-party Exporters. These metrics are then stored in a dedicated database, making Prometheus a centralized solution for metric collection in our K8s environment. The Grafana[5] tool is integrated with Prometheus to visualize these metrics. This combination is commonly used for its seamless compatibility and extensive customization, enabling comprehensive data visualization.

In addition to metric collection and visualization, our study also utilizes Elasticsearch[6], Fluentd[7], and Kibana[8] tools for log management, also known as Elastic Search, Fluentd, and Kibana (EFK) stack. Fluentd collects and filters logs from all containers and forwards them to Elasticsearch, the search engine and storage backend. Elasticsearch structures and indexes these logs, making them searchable and analyzable. Finally, Kibana is the front-end visualization tool, enabling detailed log analysis. This architecture ensures a holistic observability solution, providing real-time metrics and log analytics for effective K8s cluster management.

Machine Learning Tool: An evaluation is also conducted to determine if a ready-made ML solution can utilize anomaly detection data collected from the B5G system ob-

[2]https://projectcalico.docs.tigera.io
[3]https://openairinterface.org

[4]https://prometheus.io/
[5]https://grafana.com/
[6]https://www.elastic.co/
[7]https://www.fluentd.org
[8]https://www.elastic.co/kibana/

servability. For this goal, we used the built-in anomaly detection function in Elasticsearch. We chose the Elasticsearch function because this tool offers a robust and scalable log management and analysis solution. Moreover, Elasticsearch is open-source, providing powerful search and indexing capabilities, making it suitable for handling large volumes of log data generated by the B5G system. Elasticsearch's ready-to-use functionality allows for quicker deployment and adaptability, aligning well with the practical needs of our research. Alternative methods, such as autoencoders, require extensive training data, a more complex setup, and higher computational costs [Finke *et al.*, 2021].

This functionality operates unsupervised and requires time-series data, as represented in Figure 6. Initially, the log data extracted by Fluentd from the system is centrally stored in a single dataset in Elasticsearch (step 1). This dataset is categorized (steps 2 and 3) using an ML process that tokenizes a text field, clusteres similar data, and classifies it. This ML process is particularly suited to logs since they typically contain a finite set of possible messages. In Elasticsearch, categorization can be performed for the entire dataset or on a per-partition basis (e.g., for each service separately). In the latter case, categories are determined independently for each partition. The categorized data is stored in a secondary dataset (step 4), periodically feeding the ML model (step 5). The ML model (detector) learn each category's usual volume and pattern over time (step 6). Anomalies can be detected in two ways: (i) by observing the event rate of a particular category (count) or (ii) by identifying categories that seldom occur over time (a rare event). Finally, the analysis results are stored (step 6) for future visualization (step 7).

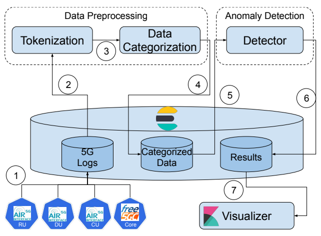

Figure 6. Unsupervised anomaly detection flow.

5 Results

This section systematically presents the outcomes of our empirical investigation, segmented into three comprehensive subsections for enhanced clarity and focus. Subsection 5.1 evaluates the effectiveness of traditional cloud-native tools in detecting anomalies within B5G system. Subsection 5.2 demonstrates the insights gained from leveraging Elasticsearch's integrated Machine Learning (ML) features for system anomaly detection. Finally, Subsection 5.3 analyzes resource usage and overhead of CNOTs.

5.1 Anomaly Detection Using Observability Tools

We aim to detect anomalies using observability tools. In this context, we realize two experiments. First, the B5G system under study is subjected to a resource provisioning failure during a connectivity test using a ping tool between a UE and the data network. Second, we submit the deployed B5G system to a resource provisioning failure during a test between a UE and the data network, stressing the environment using the iPerf tool.

Connectivity Test

In the first test scenario, the B5G system under study is subjected to a resource provisioning failure during a connectivity test between a UE and the data network. This test aims to verify if metrics and logs extracted from the system can detect the injected failure. The following steps are performed to achieve the goal: the amount of CPU provisioned for the pod running the CN services is undersized. This pod receives only half (0.25 Millicores) of the resources needed for its regular operation (0.5 Millicores). However, the RAM has sufficient resources for all pods, and RAN and CN pods are started (E1). After the initialization of the system, the Ping tool injects probes into the data session established between UE and the data network, using UE as the probe source (E2). Finally, the generation of probes is ended (E3). The metrics analyzed in this scenario are the CPU and RAM consumption of the pods corresponding to the RAN and CN services and the average latency of the sent probes. In addition, logs extracted from pods running these services are analyzed for the number of messages written per unit of time.

The contrasting outcomes between the undersized and overprovisioned scenarios in Figure 7 offer valuable insights. In the test with undersized resources, we observe a consistent maxing out of CPU consumption in the CN services pod and a threefold increase in average probe latency, compared to the overprovisioned test. These results indicate that the system is facing a bottleneck in processing capacity, severely affecting the Quality of Service (QoS). However, the overprovisioned scenario shows efficient system performance, suggesting that overprovisioning can be a viable but potentially costly strategy for maintaining system reliability.

Interestingly, the log write rates for CN and RAN services did not vary significantly between the test scenarios. This behavior suggests that CPU consumption and probe latency are effective indicators for diagnosing resource provisioning issues. However, the log write rates seem less sensitive for anomaly detection in this context. Therefore, collecting and analyzing appropriate metrics are vital for accurately detecting and diagnosing system performance and anomalies.

Performance Test

In this evaluation, we submit the deployed B5G system to a resource provisioning failure during a test between a UE and the data network. This test aims to verify whether the extracted metrics and logs analysis are suitable to detect the injected fault. To this end, we underestimate the CPU provisioned for the CN pod, receiving only a fraction (1 Millicores) of the resource required for its regular operation (4 Millicores). However, the RAM is provisioned with sufficient resources for all pods. Then, we perform the following steps. We start RAN and CN pods (E1). After initializ-

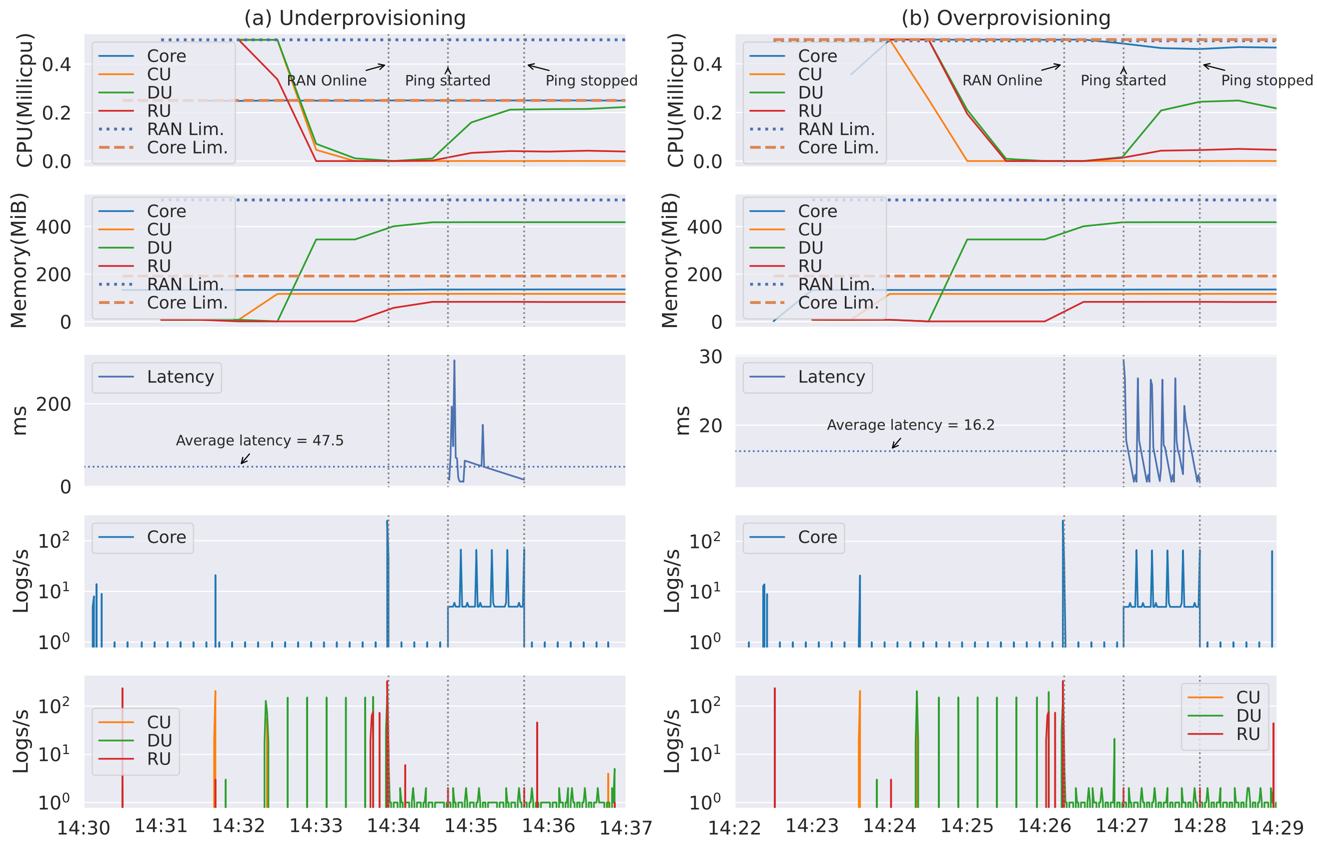

Figure 7. Results for the connectivity test scenario. The first column (a) represents the results of the undersized feature test, while the second column (b) illustrates the results of the overprovisioned resources test.

ing the system, we generate traffic between UE and the data network using the Iperf tool (E2). However, after the first minute of traffic generation, we observe that the data session established between UE and the data network is abnormally terminated (E3). Since the data session is abnormally terminated, we finish traffic generation (E4). The metrics analyzed in this scenario are the CPU and RAM consumption of the RAN and CN pods and the throughput obtained between UE and the data network.

From a comparative perspective, the results from Figure 8 (a) and (b) serve as a compelling case study for the importance of adequate resource provisioning in B5G systems. While the test with CPU underprovisioning led to an abnormal termination of the data session between UE and the data network, the overprovisioned scenario worked without any issues. This discrepancy indicates that CPU resources are a critical factor for the stability of the data sessions in this architecture. Notably, the CPU consumption in the RAN pods increased in both scenarios as traffic generation began, but only the underprovisioned test led to system instability. Therefore, it is evident that the underprovisioning of CPU resources can lead to critical system failures, affecting the user experience and overall network reliability.

Another significant observation is the timeline of anomaly detection. In the underprovisioned scenario, the DU log emitted an abnormal message even before the throughput dropped to zero. This result is especially crucial for proactive fault management. Standard cloud-native tools effectively detected the injected anomaly, but log observation provided an earlier warning than metric observation. This early detection through logs can be instrumental in triggering automated corrective actions before the user experience is severely impacted. Therefore, while metrics are valuable for understanding the system's behavior, logs can offer a more immediate insight into emerging issues, enabling quicker remediation.

5.2 Anomaly Detection Using COTS ML Tools

We use the logs extracted in the experiments described in Subsection 5.1 to evaluate whether the built-in ElasticSearch ML can detect the injected fault. To this end, we create an anomaly detection job in ElasticSearch to analyze the collected logs separately, i.e., creating one log for each pod service (CU, DU, RU, and CN). This analysis is achieved by partitioning the centralized dataset per pod name. In ElasticSearch, an anomaly detection job can be different. We use the Categorization anomaly detection, looking for categories that rarely occur in time. ML feed and processing is performed every second for each log. Figure 9 shows the screenshot of a Kibana dashboard where detected anomalies for each log are displayed for visualization. We can see that the abnormal messages emitted by the DU log are also captured by the ElasticSearch ML built-in, as illustrated by orange arrows 1, 2, 3, and 4 in Figure 9. Therefore, we argue that the deployed B5G system's logs were suitable for processing by a COTS ML solution.

Our experiments demonstrate the effectiveness of standard

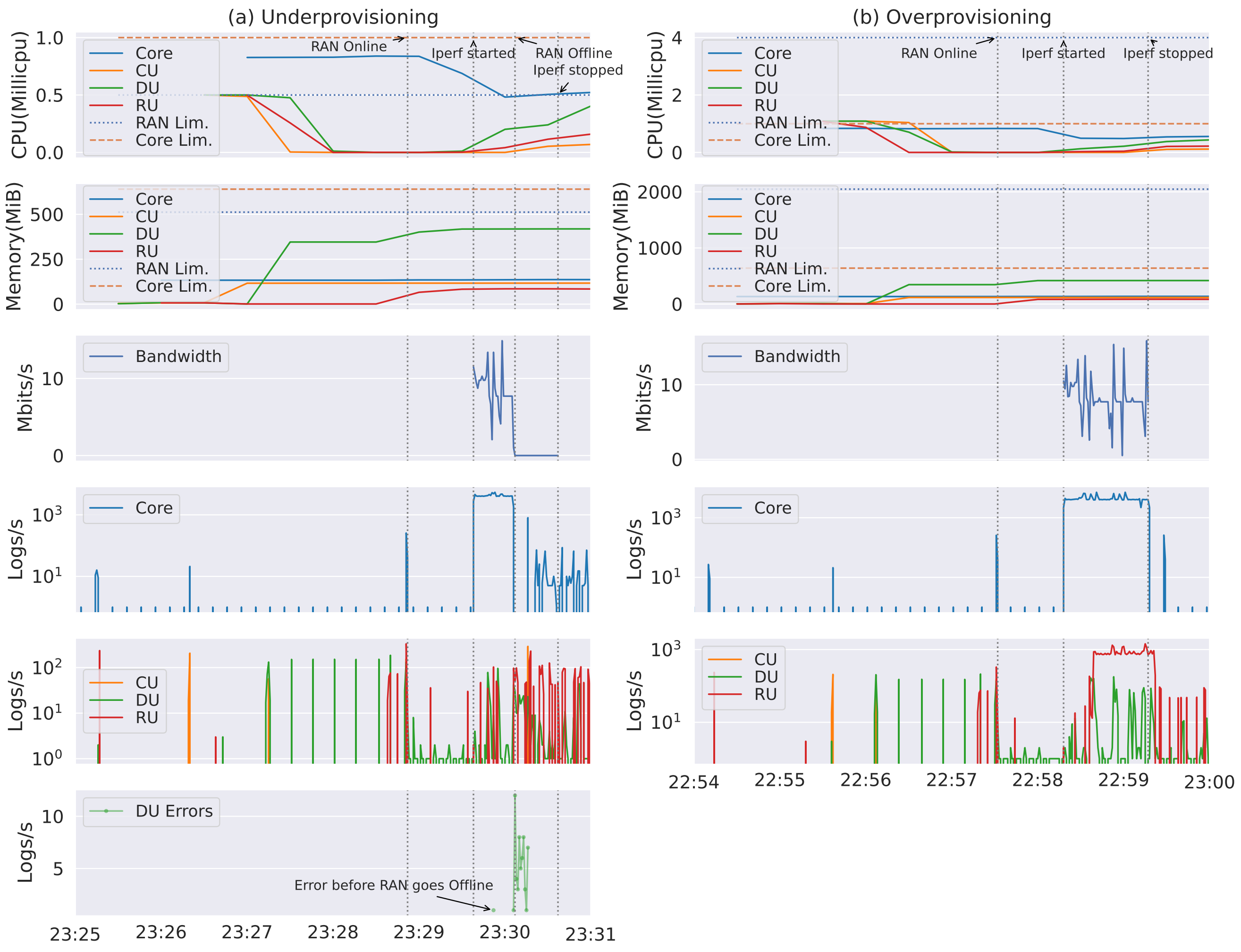

Figure 8. Results for the performance test scenario. The first column (a) represents the test results with resource underprovisioning. The second column (b) illustrates the test results with resource overprovisioning.

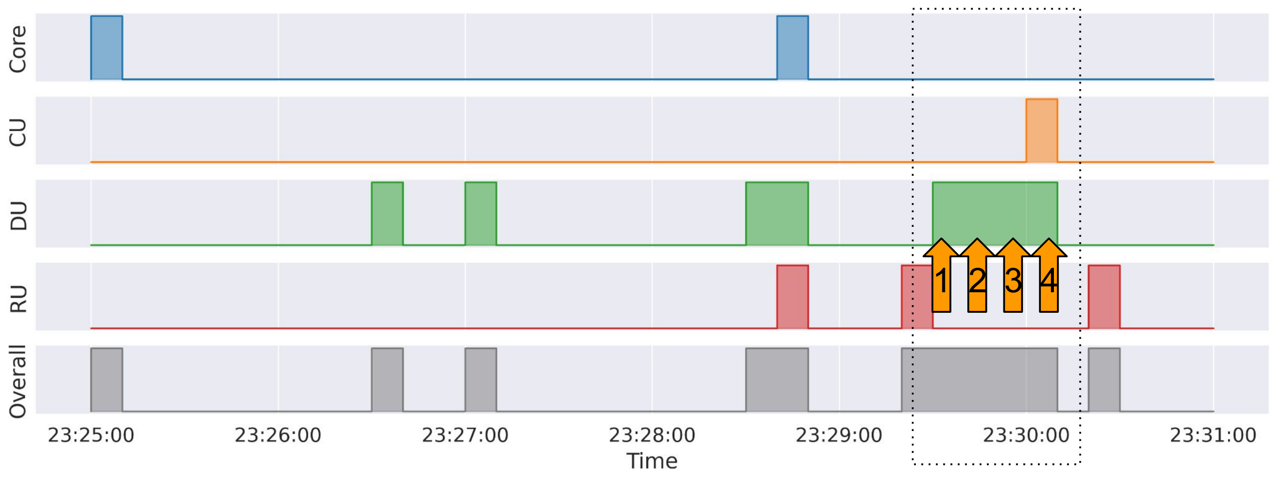

Figure 9. Anomaly detection results from Elastic Search.

CNOTs and a COTS ML solution for anomaly detection in a deployed B5G system. The ElasticSearch ML built-in was remarkably able to identify the abnormal messages in the DU logs, as highlighted in Figure 9. This result confirms the system's ability to make its internal state observable and to validate the reliability of COTS ML solutions for this purpose.

The current message logs in free5GC and OpenAirInterface could benefit further structuring and clarification. As it stands, the logs are not easily interpretable, complicating the task of visual inspection for anomalies. This observation suggests that while COTS ML solutions such as ElasticSearch can aid in anomaly detection, there is still room for enhancing the inherent observability of these B5G platforms. Improved log structuring could lead to more accurate and efficient anomaly detection, bolstering system reliability.

5.3 Overhead Analysis

Figure 10 presents the analysis of the resource utilization patterns considering CPU and RAM for K8s, Virtual Network Functions (VNFs), and CNOTs. VNFs, comprised of RAN and CN, lead in CPU resource usage, exhibiting a mean of 1.31 millicores with a standard error of 0.38. These VNFs contribute to about 57% of the total CPU overhead. The CNOT services, including Elastic Search and Kibana (ESK), Prometheus, and Fluentd, use an average CPU consumption of 2.33 millicores, approximately 22% of the total CPU. Moreover, considering CPU usage within CNOTs, ESK contributes 15%, Fluentd 2%, and Prometheus 4%. Fluentd, although critical, is among the least CPU-intensive components of the CNOT stack. In the RAM usage context, the CNOT services are notably high, consuming 86% of the system's total. Moreover, ESK consumes an average of 5.28 GB with a standard error of 0.02.

The orchestration layer, represented by K8s, presents a unique profile in the study. The orchestration uses 20% of the overall CPU and 7% of the total RAM, averaging a consumption of 0.57 GB. Given the current RAM usage, we argue that larger deployments might encounter issues. This analysis underscores the importance of an effective resource allocation strategy for efficiently accommodating K8s and other vital components in the system.

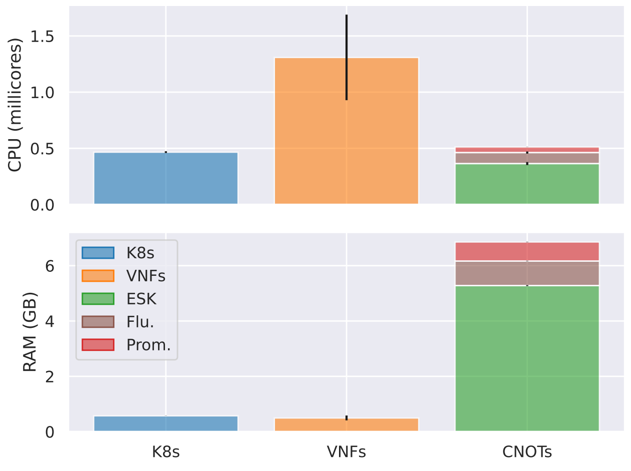

Figure 10. Resources utilization metrics.

6 Conclusion and Future Work

This empirical investigation explored the observability aspects of an open-source, cloud-native B5G system. The study demonstrates the system's compatibility with CNOTs and ML-based solutions, substantiating the potential for effective internal state monitoring. The study also exposes limitations, most notably the inadequacy of log message structuring, particularly in the RAN component. Our evaluation of the CNOT services revealed a significant memory footprint, accounting for 86% of the total memory usage and 22% overall CPU utilization.

The analysis highlights an essential requirement for more structured logging mechanisms within B5G systems, not just for clarity but also to enhance the precision and efficiency of anomaly detection methods. While metrics remain invaluable for long-term monitoring and behavior analysis, properly structured logs could offer real-time actionable insights, thus facilitating rapid issue resolution.

In future work, we will extend our study to include tracing, covering all three observability pillars. Our forthcoming research will involve a more geographically distributed setup for B5G CN and RAN to capture a robust dataset. Comparative evaluations of various open-source solutions will also be conducted to provide a comprehensive view of their capabilities. Moreover, we will integrate quantitative metrics to evaluate the effectiveness and efficiency of the selected anomaly detection tools. A focused examination of their scalability is planned due to the memory-intensive nature of CNOT services. Long-term research endeavors will probe into the role of observability within Open RAN components, including Service Management and Orchestration (SMO), Non Real-Time RAN Intelligent Controller (RIC) (non-RT RIC), and Near Real-Time RIC (near-RT RIC) [O-RAN, 2023]. Finally, a new line of investigation should address the In-band telemetry to anomaly detection in mobile networks as introduced by Fida *et al*. [2023].

Acknowledgements

This work was conducted with partial financial support from the National Council for Scientific and Technological Development (CNPq) under grant 405111/2021-5, Brazil. Additionally, the National Education and Research Network (RNP) partially supported the work through the Program PMon 2021 and resources from MCTIC, grant 01245.010604/2020-14, under the 6G Mobile Communications Systems project and by MCTIC/CGI.br/São Paulo Research Foundation (FAPESP) through projects SAMURAI - Smart B5G Core And MUltiRAn Integration under grant 2020/05127-2, Programmability, ORchestration and VIRtualization of B5G Networks (PORVIR-5G) under grant 2020/05182-3, and Slicing Future Internet Infrastructures (SFI2) under grant 2018/23097-3.

Declarations

Authors' Contributions

Competing interests

The authors declare that they have no competing interests.

Availability of data and materials

Data can be made available upon request.

Table 2. Summary of acronyms.

Acronym	Definition	Acronym	Definition
5G	fifth-generation	6G	sixth-generation
AMF	Access and Mobility Management Function	AV	Autonomous Vehicle
AUSF	Authentication Server Function	B5G	Beyond 5G
BBU	Baseband Unit	C-RAN	Cloud-RAN
CN	Core Network	CNF	Containerized Network Function
CNOT	Cloud-Native Observability Tools	COTS	Commercial-Off-The-Shelf
CPU	Central Processing Unit	CU	Central Unit
DU	Distributed Unit	DWDM	Dense Wavelength Division Multiplexing
EC	Edge Computing	EFK	Elasticsearch, Fluentd, and Kibana
ESK	Elastic Search Kibana	FaaS	Function-as-a-Service
FLML	Federate Learning ML	gNB	Next Generation Base Station
HDAD	Hybrid Deep Anomaly Detection	IoS	Internet of Senses
K8s	Kubernetes	MAC	Medium Access Control
MEC	Mobile Edge Computing	ML	Machine Learning
MPLS	Multiprotocol Label Switching	NG-RAN	Next-Generation RAN
NR	New Radio	NRF	Network Repository Function
PCF	Policy Control Function	PDCP	Packet Data Convergence Protocol
PHY	Physical Layer	POMDP	Partially Observable Markov Decision Processes
RAN	Radio Access Network	RAM	Random Access Memory
RIC	RAN Intelligent Controller	RLC	Radio Link Control
RRH	Remote Radio Head	RT	Real-Time
RU	Radio Unit	SDR	Software-Defined Radio
SMF	Session Management Function	SMO	Service Management and Orchestration
TN	Transport Network	UDM	Unified Data Management
UE	User Equipment	UPF	User Plane Function
VNF	Virtualized Network Function	VM	Virtual Machine
WAN	Wide Area Network	eBPF	extended Berkeley Packet Filter

References

3GPP (2017). Study on New Radio Access Technology; Radio Access Architecture and Interfaces (Release 14). Technical report, 3GPP. Available at: `https://www.3gpp.org/ftp//Specs/archive/38_series/38.801/38801-e00.zip`.

3GPP (2020). 5G; System architecture for the 5G System (5GS) (3GPP TS 23.501 version 16.6.0 Release 16). Technical report, 3GPP. Available at: `https://www.3gpp.org/ftp/Specs/archive/23_series/23.501/23501-gi0.zip`.

Abdulghaffar, A. *et al.* (2021). Modeling and Evaluation of Software Defined Networking Based 5G Core Network Architecture. *IEEE Access*, 9:10179–10198. DOI: 10.1109/ACCESS.2021.3049945.

Cardoso, K. V. *et al.* (2020). A Softwarized Perspective of the 5G Networks. *arXiv preprint*. DOI: 10.48550/arXiv.2006.10409.

Chen, B. and Jiang, Z. M. J. (2021). A Survey of Software Log Instrumentation. *ACM Comput. Surv.*, 54(4). DOI: 10.1145/3448976.

Doan, M. and Zhang, Z. (2020). Deep Learning in 5G Wireless Networks - Anomaly Detections. In *Proceedings of 29th Wireless and Optical Communications Conference (WOCC)*. DOI: 10.1109/WOCC48579.2020.9114924.

Duong, V. B. and Kim, Y. (2023). A Design of Service Mesh Based 5G Core Network Using Cilium. In *Proceedings of International Conference on Information Networking*, volume 2023-January, pages 25–28. IEEE Computer Society. DOI: 10.1109/ICOIN56518.2023.10049044.

Esteves, T. *et al.* (2021). CAT: Content-Aware Tracing and Analysis for Distributed Systems. In *Proceedings of the 22nd International Middleware Conference*, page 223–235. Association for Computing Machinery. DOI: 10.1145/3464298.3493396.

Fida, M., Ahmed, A. H., Dreibholz, T., Ocampo, A. F., Elmokashfi, A., and Michelinakis, F. I. (2023). Bottleneck identification in cloudified mobile networks based on distributed telemetry. *IEEE Transactions on Mobile Computing*, (01):1–18. DOI: 10.1109/TMC.2023.3312051.

Finke, T. *et al.* (2021). Autoencoders for unsupervised anomaly detection in high energy physics. *Journal of High Energy Physics*, 2021. DOI: 10.1007/JHEP06(2021)161.

Free5GC (2023). Open source 5G core network based on 3GPP R15. Available at: `https://github.com/free5gc/free5gc`. Accessed on: 2023-07-09.

Gatev, R. (2021). *Observability: Logs, Metrics, and Traces*, pages 233–252. Apress, Berkeley, CA. DOI: 10.1007/978-1-4842-6998-5.

Gizchina (2021). The number of 5G users worldwide will reach 2.6 billion in five years. Available at:`https://www.gizchina.com`. Accessed: 2023-09-09.

Gomez Blanco, D. (2023). *Adopting Open Observability Standards Across Your Organization*, pages 217–229. Apress, Berkeley, CA. DOI: 10.1007/978-1-4842-9075-0.

Hakiri, A. *et al.* (2022). Techniques for Realizing Secure, Resilient and Differentiated 5G Operations. In *Proceedings of 14th IFIP Wireless and Mobile Networking Conference (WMNC)*, pages 113–117. DOI: 10.23919/WMNC56391.2022.9954310.

Hung, M.-H. *et al.* (2022). A SDN Controller Monitoring Architecture for 5G Backhaul Networks. In *Proceedings of 23rd Asia-Pacific Network Operations and Management Symposium (APNOMS)*, pages 1–4. DOI: 10.23919/APNOMS56106.2022.9919988.

Joda, R. *et al.* (2022). The internet of senses: Building on semantic communications and edge intelligence. *IEEE Network*, pages 1–9. DOI: 10.1109/MNET.107.2100627.

John, W. *et al.* (2017). Meeting the Observability Shallenges for VNFs in 5G Systems. In *IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, pages 1127–1130. DOI: 10.23919/INM.2017.7987445.

Kim, H. J. *et al.* (2022). Network Anomaly Detection based on Domain Adaptation for 5G Network Security. In *Proceedings of International Conference on ICT Convergence*, volume 2022-October, pages 976–980. IEEE Computer Society. DOI: 10.1109/ICTC55196.2022.9952454.

Klinkowski, M. and Jaworski, M. (2022). Planning of Optical Connections in 5G Packet-Optical xHaul Access Network. *Applied Sciences*, 12(3). DOI: 10.3390/app12031146.

Larsen, L. M. P., Checko, A., and Christiansen, H. L. (2019). A Survey of the Functional Splits Proposed for 5G Mobile Crosshaul Networks. *IEEE Communications Surveys Tutorials*, 21(1):146–172. DOI: 10.1109/COMST.2018.2868805.

Leliopoulos, P. and Drigas, A. (2023). Big data and data analytics in 5G mobile networks. *Global Journal of Engineering and Technology Advances*, 15:165–190. DOI: 10.30574/gjeta.2023.15.3.0114.

Li, B. *et al.* (2022). Enjoy your observability: an industrial survey of microservice tracing and analysis. *Empirical Software Engineering*, 27:1–28. DOI: 10.1007/s10664-021-10063-9.

O-RAN (2023). O-RAN ALLIANCE. Available at: : https://www.o-ran.org. Accessed on em 01/02/2023.

OpenAirInterface (2023). Openairinterface 5G Wireless Implementation. Available at:`https://gitlab.eurecom.fr/oai/openairinterface5g/`. Accessed on:2023-07-09.

Polese, M. *et al.* (2023). Understanding o-ran: Architecture, interfaces, algorithms, security, and research challenges. *Commun. Surveys Tuts.*, 25(2):1376–1411. DOI: 10.1109/COMST.2023.3239220.

Prathiba, S. B. *et al.* (2022). A Hybrid Deep Sensor Anomaly Detection for Autonomous Vehicles in 6G-V2X Environment. *IEEE Transactions on Network Science and Engineering*. DOI: 10.1109/TNSE.2022.3188304.

Raja, G. *et al.* (2022). AI-Empowered Trajectory Anomaly Detection and Classification in 6G-V2X. *IEEE Transactions on Intelligent Transportation Systems*. DOI: 10.1109/TITS.2022.3197446.

Saavedra, A. *et al.* (2018). WizHaul: On the Centralization Degree of Cloud RAN Next Generation Fronthaul. *IEEE Transactions on Mobile Computing*, 17(10):2452–2466. DOI: 10.1109/TMC.2018.2793859.

Scrocca, M. *et al.* (2020). The Kaiju Project: Enabling Event-Driven Observability. *14th ACM International Conference on Distributed and Event-Based Systems*, pages 84–91. DOI: 10.1145/3401025.3401740.

Soldani, D. *et al.* (2023). eBPF: A New Approach to Cloud-Native Observability, Networking and Security for Current (5G) and Future Mobile Networks (6G and Beyond). *IEEE Access*. DOI: 10.1109/ACCESS.2023.3281480.

Statista (2023). 5G Networks Deployment World Map. Available at: `https://www.statista.com/chart/23194/5g-networks-deployment-world-map/`. Accessed: 2023-09-09.

Sun, P. *et al.* (2020). Adaptive Rule Engine for Anomaly Detection in 5G Mobile Edge Computing. In *Proceedings of the IEEE 20th International Conference on Software Quality, Reliability, and Security (QRS-C)*, pages 690–691. Institute of Electrical and Electronics Engineers Inc.. DOI: 10.1109/QRS-C51114.2020.00123.

Surantha, N. and Putra, N. A. (2022). Integrated SDN-NFV 5G Network Performance and Management-Complexity Evaluation. *Future Internet*, 14. DOI: 10.3390/fi14120378.

Tang, Q. *et al.* (2022). A Systematic Analysis of 5G Networks With a Focus on 5G Core Security. *IEEE Access*, 10:18298–18319. DOI: 10.1109/ACCESS.2022.3151000.

Yuan, Y. *et al.* (2022). Insight of Anomaly Detection with NWDAF in 5G. In *Proceedings of the International Conference on Computer, Information and Telecommunication Systems (CITS)*. Institute of Electrical and Electronics Engineers Inc.. DOI: 10.1109/CITS55221.2022.9832914.

Zheng, Y. *et al.* (2022). Mahalanobis Distance and Pauta Criterion based Log Anomaly Detection Algorithm for 5G Mobile Network. In *Proceedings of the IEEE 21st International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, pages 1384–1389. DOI: 10.1109/TrustCom56396.2022.00195.