TimeTrack: A Dataset for Exploring Temporal Patterns and Predictive Insights into OpenAirInterface (OAI) CI/CD Cluster

Meliani Abd Elghani
Communication systems
Eurecom
Biot, France
meliani@eurecom.fr

Arora Sagar
OpenAirInterface Alliance
Biot, France
sagar.arora@openairinterface.org

Adlen Ksentini
Communication systems
Eurecom
Biot, France
ksentini@eurecom.fr

Raymond Knopp
communication systems
Eurecom
Biot, France
knopp@eurecom.fr

***Abstract*—The rapid adoption of cloud-native architectures and telecommunications frameworks has intensified the need for efficient resource management and network performance monitoring. Time series data plays a critical role in these settings by capturing sequential changes in metrics, enabling advanced analytics and machine learning models for anomaly detection, predictive maintenance, and resource allocation optimization. However, existing public datasets lack the granularity or breadth of metrics required for in-depth research, often omitting critical computing and network performance indicators, particularly within cloud-native environments for telecommunications workloads. TimeTrack addresses this gap by offering a comprehensive time series dataset collected from an OpenAirInterface (OAI) Continuous Integration/Continuous Deployment (CI/CD) cluster. The dataset contains monitoring data on both computing and network resources used by the CI/CD of OAI components (gNodeB, User Equipments and Core Network), providing a unique perspective on the demands of cloud-native telecommunications systems. The primary objective of TimeTrack is to support research in resource management, anomaly detection, and network optimization in similar environments.**

***Index Terms*—Time Series, Dataset, Monitoring, Computing resources, Network Resources**

I. Introduction

The rapid adoption of cloud-native architectures and telecommunications frameworks has intensified the need for efficient resource management and network performance monitoring [1]. Time series data, in particular, plays a critical role in such settings by capturing sequential changes in metrics, enabling the development of advanced analytics and machine learning models for anomaly detection, predictive maintenance, and resource allocation optimization. Existing public datasets often fall short in this regard, lacking the granularity or breadth of metrics required for in-depth research. They frequently omit either computing resources (e.g., CPU, memory, disk bandwidth) or network performance indicators (e.g., latency, jitter, packet loss) within cloud-native environments, especially in the context of telecommunications workloads. This gap limits researchers from training and validating machine learning models that rely on a holistic view of system performance, making it challenging to create algorithms capable of accurate forecasting and anomaly detection in dynamic, real-world settings. To address this need, we present ***TimeTrack***, a comprehensive time series dataset collected from one of the OAI [2] test clusters generally used to run CI/CD workloads [3]. TimeTrack provides continuous monitoring of both computing and network resources, offering a unique perspective on the fluctuating demands of cloud-native telecommunications systems. The primary objective of this dataset is to support research in resource management, machine learning-based anomaly detection, and network optimization in similar environments We have made TimeTrack publicly available at [4]. The contributions of this work are:

- **Comprehensive Dataset:** TimeTrack captures essential infrastructure metrics across both computing and network domains, facilitating nuanced analyses for a diverse range of performance optimization and predictive modeling applications.
- **Support for Machine Learning and Advanced Analytics:** The dataset's high granularity makes it highly effective for use in machine learning pipelines, where it supports the training of models for anomaly detection, forecasting, and predictive maintenance.
- **Real-world Applicability:** Collected from an operational environment, TimeTrack offers data that accurately represents real-world workloads, making it an ideal resource for benchmarking models and testing algorithms in both cloud and telecommunications research.

In the following sections, we review related work, outline the data collection process, and provide an overview of the dataset structure. We then analyze the collected data, discussing potential applications and limitations, and demonstrate TimeTrack's value for training machine learning models. Finally, we conclude with a summary of our findings and explore the dataset's broader implications.

II. Related Work

Several publicly accessible datasets exist, such as Google Cluster Data 2011 [5], Alibaba Microservices Traces 2022 [6], Grid Workloads Archive GWA-T-13-Materna [7], and

Azure public traces v1 [8] and v2 [9], which offer valuable insights into resource utilization in distributed computing environments. However, these datasets have limitations that may reduce their effectiveness for certain research purposes. First, these datasets either span only a few days, limiting their ability to reveal long-term trends and patterns in resource utilization, or they focus solely on computing metrics, neglecting other important aspects such as network performance or storage metrics, which are crucial for comprehensive analysis. Additionally, some datasets, particularly those from older grid computing systems, may not align well with current cloud computing environments, making them less relevant for contemporary research. Furthermore, these datasets frequently lack detailed information about the workloads running on the machines, such as specific objectives, constraints, or task nature. This missing context is important for analyses like workload characterization or the development of predictive models. Another notable limitation is that the data is often recorded at relatively coarse intervals, such as every 5 minutes. While this frequency might suffice for general monitoring, it poses challenges for applications such as time series forecasting, which require finer-grained data to build accurate models. The 5-minute interval is particularly inadequate for capturing rapid fluctuations in resource usage or for training models that depend on high-frequency data for precise predictions.

Moreover, most related works rely on data collected in virtualized environments, where resource usage metrics can be impacted by virtualization overhead and may not fully reflect actual hardware performance. In contrast, TimeTrack was collected on physical machines, eliminating virtualization artifacts and providing a more accurate representation of resource utilization, network performance, and system behavior. This setup allows TimeTrack to capture real-world performance metrics and transient behaviors that are especially important for high-precision applications such as machine learning model training and anomaly detection, where exact resource usage patterns are crucial for building effective predictive models. Finally, while some of these datasets provide only general resource usage information, TimeTrack delivers highly detailed metrics across computing and networking domains. This detailed structure will be presented in the next section, and Table II offers a comparison between TimeTrack and the related works discussed in this section.

III. TimeTrack

A. Environment, Data Collection and Data Structure

Our data was collected at 45-second intervals over a period of one month from an OAI testing cluster, which is primarily used for running CI/CD pipelines for the development of various OAI 5G network components, such as gNB, UE, and CN. The cluster consists of seven machines with a total of 437.5 GB of RAM, 236 CPU cores, approximately 1800 GB of SSD storage, and 38 physical network interfaces, this number represents maximum number of network interfaces present in the cluster (Ethernet/Fiber), however, only 3 are used for each machine (2 Ethernet and 1 fiber interfaces). The data was collected using Prometheus as the primary metric source, with the Prometheus plugin serving as an intermediary between Prometheus and the collector component. The distribution of resources across the machines is detailed in Table I.

TABLE I
Resource Distribution Across Machines

Machine (No)	Cores	RAM(GB)	Disk(GB)	Physical(No)
1	36	62.5	278.37	4
2	48	62.5	222.5	6
3	36	62.5	278.37	6
4	36	62.5	278.37	4
5	24	62.5	222.5	6
6	36	62.5	278.37	6
7	20	62.5	222.5	6

The dataset contains four essential traces:

1) **Compute Metrics:** This trace includes the available and used memory amounts at both the cluster level and across individual machines. It also records the average CPU availability and consumption across all machines, along with the used and remaining disk space at both the cluster and machine levels. Additionally, this trace captures the read/write disk throughputs for each machine. Given the large number of CPU cores, the detailed utilization and availability of each core are provided in a separate trace.
2) **CPU Core Utilization:** This separate trace provides detailed information on the utilization and availability of each individual CPU core, effectively addressing the complexity arising from the large number of cores in the cluster.
3) **Network Latency Metrics:** This trace measures the minimum, maximum, average, and mean deviation (mdev) of round-trip time (RTT) and jitter between the OAI setup and Google's DNS server (8.8.8.8).
4) **Network Interface Metrics:** The final trace captures the percentage of dropped and error network packets, along with the transmitted and received throughputs for the physical network interfaces.

B. Analysis

Figure 1 presents the correlation matrix for the compute metrics across the cluster, where "CU" represents CPU usage, "UM" represents memory usage, "DRT" denotes disk read throughput, "DWT" stands for disk write throughput, and "UD" indicates disk usage.

As illustrated, there is a strong positive correlation between the cluster's overall memory usage and the memory usage on individual machines. This correlation is expected because the cluster's memory usage is essentially the sum of the memory consumed by all the machines. As each machine increases its memory consumption, the cluster's total memory usage correspondingly rises, resulting in this high positive correlation. Furthermore, the correlation matrix reveals a notable positive correlation between the percentage of CPU usage and the amount of memory used on several machines, specifically

TABLE II
COMPARISON OF TIMETRACK WITH RELATED DATASETS

Dataset	Number Of machines	Collection time Interval	Duration	Setup	Detail Level	Collected metrics
[5]	12500	5 min	29 days	Virtual	medium	Compute
[6]	10000	5 min	13 days	Physical	low	Compute
[7]	1594	5 min	3 months	Virtual	medium	Compute & Network & Storage
[8]	2,013,767	5 min	30 days	Virtual	low	Compute
[9]	2,695,548	5 min	30 days	Virtual	medium	Compute
TimeTrack	7	45 sec	30 Days	Physical	high	Compute & Network & Storage

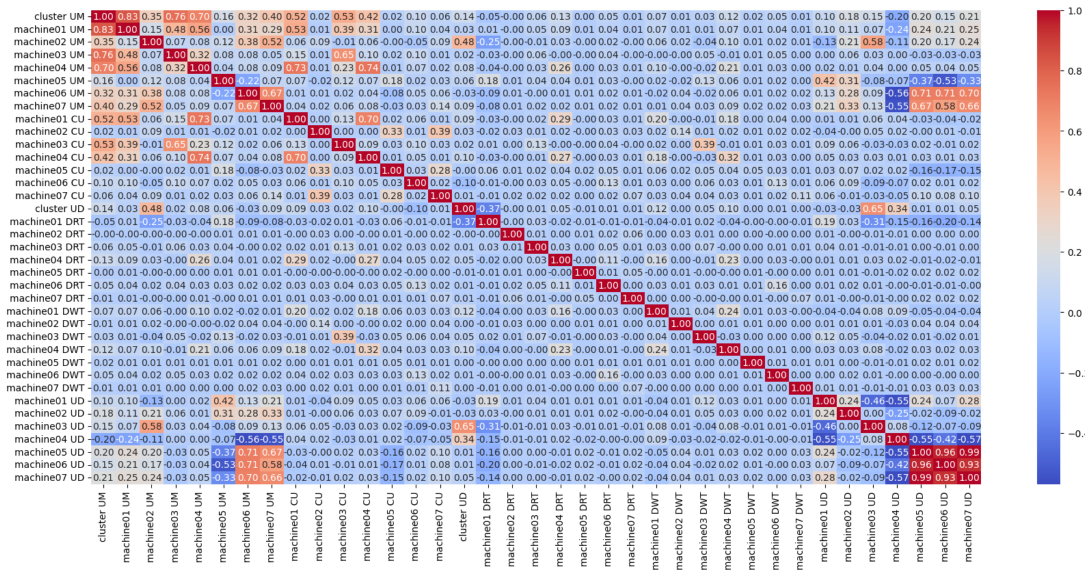

Fig. 1. Correlation matrix of compute metrics across the cluster. The matrix shows the relationships between CPU usage (CU), memory usage (UM), disk read throughput (DRT), disk write throughput (DWT), and disk usage (UD) at both the cluster and individual machine levels.

0.53 for machine 01, *0.65* for machine 03, *0.74* for machine 04, and *0.7* for machine 07. This pattern suggests that as CPU usage increases, there is often a corresponding rise in memory consumption, which could indicate that many of the workloads running on these machines are both CPU and memory-intensive. However, this relationship can vary depending on the nature of the workloads. Figure 1 also shows that some machines, specifically machine 06 and machine 07, exhibit a strong correlation between memory usage and disk usage. One potential reason for this correlation is related to the nature of the workloads running on these machines. In many cases, applications that handle large datasets or perform intensive data processing operations may load significant amounts of data into memory for faster access. However, to ensure data persistence, these applications might simultaneously write large volumes of data to disk, particularly if the data needs to be saved or logged frequently.

Furthermore, applications that use memory-mapped files—where the contents of files on disk are mapped directly into the address space of a process—could also explain this correlation. In such scenarios, as memory usage increases due to the mapping of more file data into memory, there could be a corresponding increase in disk activity, leading to the observed correlation between memory and disk usage.

Other metrics, such as disk write throughput (DWT) and disk read throughput (DRT), show no significant correlation with each other or with other metrics, as seen in Figure 1. However, a particularly notable observation is that the overall disk usage at the cluster level does not exhibit a strong correlation with the disk usage on individual machines. This prompted us to further investigate by plotting the disk usage values for both the entire cluster and the individual computing nodes, as shown in Figure 2.

Upon examining these plots, it became evident that many machines follow a routine of deleting data at varying intervals. This behavior is expected, as the cluster is primarily used for running CI/CD pipelines, which can generate a substantial amount of temporary data, such as unnecessary software builds. These builds are often removed after their utility has passed, explaining the lack of correlation between the cluster's overall disk usage and the disk usage on individual machines.

Figure 3 shows the average CPU usage (CU), memory usage (MU), and disk usage (DU) per machine. Except for machine 06, resource utilization is generally consistent across the machines, indicating effective load distribution and balanced computing resource sharing across the cluster.

Figure 4 displays two time series plots. The top plot represents the average RTT to Google DNS (8.8.8.8) throughout

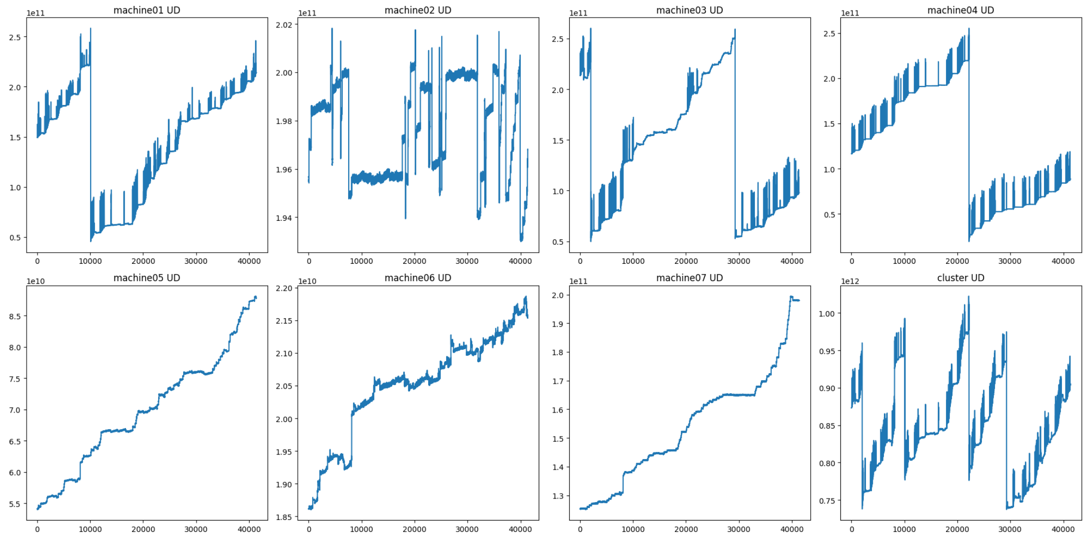

Fig. 2. disk usage plots for the entire cluster and individual nodes

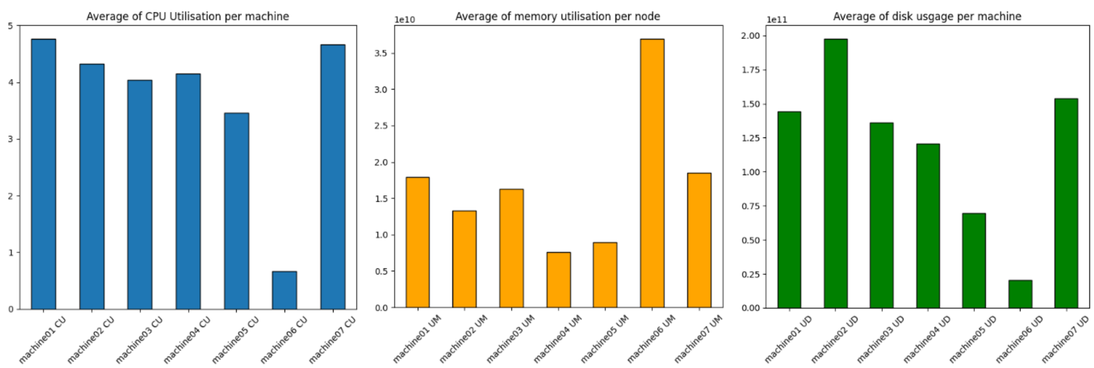

Fig. 3. Average CPU usage (CU), memory usage (MU), and disk usage (DU) per machine in the cluster.

the monitoring period, measured in milliseconds. The bottom plot shows the jitter values for the same connection, also in milliseconds. We observe significant fluctuations in RTT over time, with peaks surpassing 80 ms. These spikes may be indicative of network congestion or changes in the network path between the OAI setup and Google DNS. However, most RTT values remain within the 0-10 ms range, which suggests generally stable connectivity during the majority of the monitoring period.

The jitter plot reveals variability in packet delay, an essential metric for assessing network performance, especially in real-time applications. Similar to RTT, jitter exhibits considerable spikes, with some values reaching up to 50 ms. Such high jitter can lead to inconsistencies in network performance, particularly for services that require stable and consistent timing. A notable observation is the correlation between the two plots; spikes in RTT often coincide with increases in jitter. This suggests that the same underlying factors contributing to higher RTT, such as network congestion or path variations, are also causing increased jitter, highlighting the interconnected nature of these network performance metrics.

Figure 5 illustrates the memory utilization across the cluster over a week, highlighting distinct patterns between weekend and weekday usage. During the weekend (first two plots), memory usage remains low and stable, likely due to minimal active workloads, reflecting an idle or background operational state. In contrast, on weekdays (last five plots), memory utilization rises significantly during working hours, suggesting high demand from active workloads before dropping to a stable baseline after hours. This regular daily cycle indicates predictable demand patterns, which can support efficient resource management strategies. Similar trends observed in CPU and disk usage further reinforce this consistency, enabling the potential for dynamic scaling and optimized resource allocation based on these workload cycles.

IV. Evaluation

A. Experiments

To assess the quality of our dataset compared to other states of the datasets, we used our data and the GWA Materna [7] traces to train models aimed at forecasting CPU consumption. We chose CPU consumption as the key metric because it

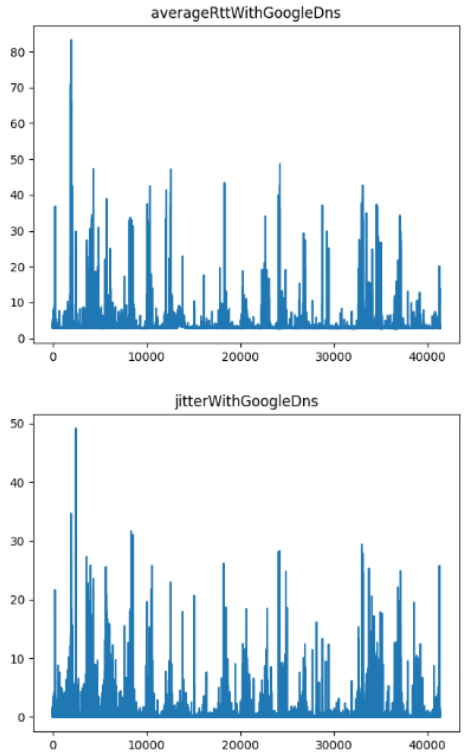

Fig. 4. Time series plots showing network performance metrics between the OAI setup and Google DNS (8.8.8.8)

directly impacts workload management and system stability in distributed environments, which makes it a critical factor in performance forecasting. For the models, we selected widely-used architectures in time series forecasting—LSTM, RNN, GRU, and CNN—as they capture temporal patterns effectively. LSTMs, RNNs, and GRUs are well-suited for sequential data due to their recurrent structures, while CNNs can capture local patterns efficiently, adding diversity to the comparison. We varied the time window—a parameter that determines the number of past time steps each model observes at each training instance—from 1 to 20. This allowed us to understand how each dataset performed under different forecasting ranges. We specifically selected the GWA Materna traces from other state-of-the-art datasets due to its structured, machine-level data organization. The traces offer convenient data access, with around 8000 time series values for each machine, making it ideal for controlled experiments. For this experiment, we randomly selected a file from the GWA Materna traces representing the metrics of one virtual machine and extracted the first 8000 CPU time series values. Similarly, we used the first 8000 CPU values from our dataset's "machine 01" in the Timetrack trace. Finally, to evaluate these models, we used real-time data from a local bare-metal Kubernetes cluster, provisioned specifically for experimental purposes. The test values were fed in 30-second intervals, allowing us to test the models against live CPU consumption patterns.

B. Results

To evaluate the results obtained from training on our dataset versus the GWA Materna traces, we focused on key time series model metrics: Mean Squared Error (MSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE). These metrics are crucial in time series forecasting as they provide insight into the accuracy and stability of model predictions. MSE penalizes larger errors more heavily, making it ideal for highlighting significant prediction deviations. MAE, on the other hand, gives a straightforward average error value, while MAPE provides a percentage-based error measure, making it easier to interpret across different scales.

Figure 6 presents plots of key metrics—MSE, MAE, and MAPE—across different model types (LSTM, RNN, GRU, CNN) as a function of time window size. The results display a consistent trend in which models trained on the Timetrack dataset (in blue) generally outperform those trained on the GWA Materna data (in red). This performance difference suggests that models trained on the Timetrack dataset demonstrate better generalization and more stable predictions, indicating a higher data quality for predictive tasks. A primary reason for this disparity is that the Timetrack dataset was collected in real-world environments, yielding data that captures natural fluctuations in resource consumption more effectively. Additionally, the Timetrack data was collected at relatively short intervals (45 seconds), allowing models to capture transient spikes and dips in CPU usage with greater accuracy. In contrast, the GWA Materna data, sampled at 5-minute intervals, likely misses these short-term variations in CPU consumption. This limitation can result in less consistent model performance on GWA data, as it lacks the fine-grained insights necessary to respond accurately to rapid changes in the time series data.

V. Conclusion

The TimeTrack dataset offers an extensive collection of time series data from a real-world cloud-native environment, with a focus on both computing and network metrics. Experimental results demonstrate that models trained on TimeTrack yield more accurate and stable predictions compared to those trained on similar datasets, such as the GWA Materna traces, especially in scenarios requiring high-frequency data. This advantage arises from TimeTrack's fine-grained collection intervals, which enable models to capture transient fluctuations in resource usage, making it ideal for predictive maintenance and resource allocation tasks. The dataset addresses a critical gap for researchers working on real-time, high-precision applications, enabling a comprehensive approach to performance monitoring and optimization in cloud-native environments.

Acknowledgment

This is supported by the European Union's Horizon Research Innovation program AC^3 (grant agreement No 101093129), SLICES-PP Coordinated Support Action (grant agreement ID 101079774), and FRANCE 2030 Future Networks Experimental Research Platform Programme (PEPR).

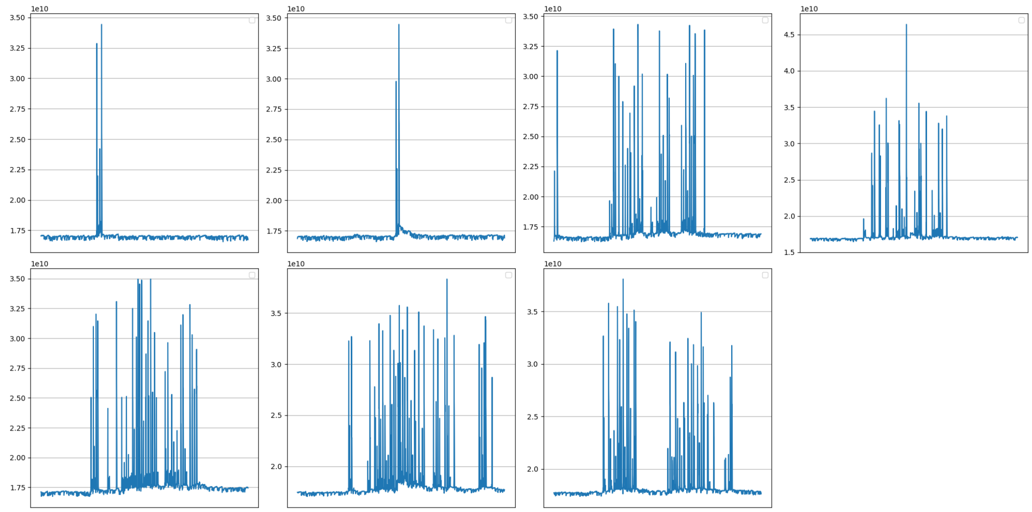

Fig. 5. Memory utilization across the cluster over a week. showing low and stable usage during the weekend (first two plots) and increased activity during weekday working hours (last five plots).

Fig. 6. Comparison of forecasting performance metrics (MSE, MAE, MAPE) across models (LSTM, GRU, RNN, CNN) trained on the Timetrack dataset (blue) and the GWA Materna traces (red) with varying time window sizes.

REFERENCES

[1] B. Brik and A. Ksentini, "On predicting service-oriented network slices performances in 5g: A federated learning approach," in *2020 IEEE 45th Conference on Local Computer Networks (LCN)*, 2020, pp. 164–171.

[2] OpenAirInterface, "OpenAirInterface: 5G Software Alliance for Democratising Wireless Innovation," 2023, accessed: 2024-10-30. [Online]. Available: https://openairinterface.org/

[3] ——, "OpenAirInterface Ci/Cd webpage," 2023, accessed: 2024-10-30. [Online]. Available: https://openairinterface.org/test-measurement/

[4] A. Meliani, "Timetrack: Openairinterface (oai) ci/cd time series dataset," 2024, accessed: Feb 13, 2025. [Online]. Available: https://www.kaggle.com/datasets/abdelghanimeliani/open-air-interface-cicd-timeseries-dataset/data

[5] Google Inc., "Google Cluster Data 2019," 2019, accessed: 2024-08-27. [Online]. Available: https://github.com/google/cluster-data/blob/master

[6] Alibaba Inc., "Alibaba Cluster Trace v2022," 2018, accessed: 2024-08-27. [Online]. Available: https://github.com/alibaba/clusterdata/blob/master/cluster-trace-microservices-v2022

[7] Delft University of Technology, "Grid Workloads Archive GWA-T-13-Materna," accessed: 2024-08-27. [Online]. Available: http://gwa.ewi.tudelft.nl/datasets/gwa-t-13-materna

[8] Microsoft Corporation, "Microsoft Azure Public Dataset V1," 2019, accessed: 2024-10-30. [Online]. Available: https://github.com/Azure/AzurePublicDataset/blob/master/

[9] ——, "Microsoft Azure Public Dataset V2," 2020, accessed: 2024-10-30. [Online]. Available: https://github.com/Azure/AzurePublicDataset/blob/master/