When MLOps Meets NWDAF to Enable Autonomous Next Generation Network Analytics

Mazene Ameur*, *Member, IEEE*, Bouziane Brik‡, *Senior Member, IEEE,* and Adlen Ksentini*, *Senior Member, IEEE*

* Communication Systems Department, EURECOM, Sophia Antipolis, France.

‡ Computer Science Department, University of Sharjah, Sharjah, UAE.

Emails: {firstname.lastname}@eurecom.fr, bbrik@sharjah.ac.ae

***Abstract*—The shift toward 6G intensifies the need for Zero-Touch Service Management (ZSM) to enable autonomous network operations. Central to this vision is the Network Data Analytics Function (NWDAF), a standardized entity defined by the 3rd Generation Partnership Project (3GPP) to deliver data-driven intelligence within the 5G core network. While 3GPP Release 17 introduced the architectural separation of NWDAF into the Analytics Logical Function (AnLF) and the Model Training Logical Function (MTLF), the specifications leave the model lifecycle management largely undefined. This creates a significant specification-to-implementation gap that hinders truly autonomous analytics. Furthermore, existing NWDAF open-source initiatives lack coherent ML Operations (MLOps) integration, particularly for automated model training and lifecycle orchestration. This paper addresses these limitations by presenting the first comprehensive, standards-aligned open-source NWDAF implementation fully compliant with 3GPP Release 17. Our contribution extends the OpenAirInterface NWDAF by establishing the AnLF–MTLF architectural separation with an MLOps-enabled MTLF that autonomously orchestrates the model lifecycle. The proposed NWDAF is validated across representative 3GPP use cases, demonstrating stable autonomous model training, efficient end-to-end analytics provisioning, and consistent predictive performance with low inference latency.**

***Index Terms*—NWDAF, 3GPP, Release 17, OpenAirInterface, MLOps, ZSM, Open-source.**

I. Introduction

The advent of fifth-generation (5G) networks has enabled mission-critical industrial applications (e.g., Industry 5.0 and autonomous driving). However, as we transition to sixth-generation (6G) networks, these applications demand far stricter requirements on reliability, latency, and dynamic resource allocation, rendering traditional rule-based management obsolete [1]. This massive complexity mandates a shift towards Zero-Touch Service Management (ZSM), where Artificial Intelligence (AI) and Machine Learning (ML) autonomously optimize performance by extracting insights from network telemetry, thereby efficiently managing the dynamic service flow [2]. To support this paradigm shift, the 3rd Generation Partnership Project (3GPP) introduced the Network Data Analytics Function (NWDAF) as a native component of the 5G core's Service-Based Architecture (SBA) [3]. First specified in Release 15, NWDAF provides standardized analytics capabilities by collecting network data, applying static ML-based techniques, and generating predictive insights for consumer Network Functions (NFs). Release 17 significantly advanced this framework by architecturally separating the Model Training Logical Function (MTLF), responsible for model training and management, from the Analytics Logical Function (AnLF), dedicated to inference and analytics delivery [4]. This modular decomposition, coupled with standardized service exposure interfaces (MLModelTraining and MLModelProvision), enables seamless ML operations and allows NWDAF consumers to participate more directly in the analytics ecosystem, firmly establishing NWDAF as a cornerstone of autonomous core network intelligence.

Meanwhile, recent and rapid progress in Machine Learning Operations (MLOps) has emerged as the essential engineering discipline for industrializing AI-driven systems. MLOps provides a systematic, automated approach to manage the entire ML lifecycle, spanning data validation, model training orchestration, versioning, continuous deployment, and iterative monitoring and retraining [5]. By providing tools and processes to automate and govern these steps, MLOps transforms ad-hoc development into reliable pipelines that preserve model quality, maintain operational stability, and support autonomous network management.

Despite the standardization efforts, significant gaps still impede truly autonomous NWDAF deployments. While Release 17 defines the functional AnLF–MTLF separation and the associated service interfaces, the specifications leave model-lifecycle operations out of scope, including training orchestration, versioning, deployment automation, and continuous retraining. This specification-to-implementation gap forces ad-hoc design decisions without interoperability guarantees. Moreover, existing open-source NWDAF initiatives provide only partial compliance with 3GPP requirements and lack a systematic MLOps backbone. Current implementations do not support automated or continuous model-training pipelines, nor can they orchestrate coordinated model-lifecycle operations across the AnLF, MTLF, and consumer NFs. These limitations hinder the adoption of NWDAF as a practical vehicle for intelligent network automation.

To address these limitations, this paper presents a mature, standards-compliant NWDAF implementation integrating advanced ML capabilities. **To the best of our knowledge,** this is the first open-source implementation leveraging an end-to-end service-based MLOps platform natively for MTLF and integrated with AnLF and OpenAirInterface (OAI) 5G Core

Network.

To this end, the main contributions of this paper can be summarized as follows:

- We present the first open-source, 3GPP Release 17-compliant NWDAF implementation featuring full separation of AnLF and MTLF, addressing a critical gap in existing open-source initiatives.
- We bridge the standardization-implementation gap by designing and implementing a service-based MLOps architecture for MTLF, explicitly addressing model lifecycle management aspects left undefined in 3GPP specifications, including model registry, artifact storage, and metadata management.
- We implement standards-compliant service exposure interfaces (i.e., MLModelTraining and MLModelProvision) for the MTLF with Microservice-based integration, enabling seamless interoperability among the MTLF, AnLF, and consumer NFs.
- We comprehensively validate our NWDAF implementation across key 3GPP use cases, demonstrating an autonomous training and provisioning pipeline tailored to support next-generation ZSM autonomous networks.
- To promote reproducibility and support the open-source community, we publicly release a mature, standards-aligned NWDAF implementation[1] to accelerate AI/ML-driven network automation research for 5G and 6G.

The remainder of this paper is structured as follows. Section II reviews existing NWDAF standards and implementations and identifies the key gaps. Section III details the proposed NWDAF architectural design. Section IV describes the implementation and experimental setup and provides a comprehensive analysis of the results. Finally, Section V summarizes the main findings and outlines future research directions.

II. Related Work

Recent studies position NWDAF as a central enabler of data-driven automation in emerging 5G networks, with work spanning monitoring, analytics generation, and ML-assisted decision making. In [6], NWDAF was integrated with an intent-based networking framework to automate network slice lifecycle management, but the design relied on the OAI Evolved Packet Core (OAI EPC), which lacks standardized 5G core functions and exposure services. Bayleyegn *et al.* [7] implemented an NWDAF on free5GC using specific exposure services to generate mobility and communication load analytics using classical regression models trained on packet-trace data. However, their system omits User Equipment (UE) communication insights from the User Plane Function (UPF) exposure service, limiting its applicability for analytics that require full user-plane visibility.

The authors in [8] proposed an NWDAF integrated with the Open5GS core and implemented service-based interfaces, including N34 for the Network Slice Selection Function (NSSF) and N23 for the Policy Control Function (PCF), enabling

[1]https://gitlab.eurecom.fr/netsoft/nwdaf-rel17

TABLE I: Related Work Analysis.

Work	Core Network Stack	Service Exposure	MLModel Provision API	MLModel Training API	MLOps Stack
[8]	Open5GS	✗	✗	✗	✗
[9]	Open5GS	✗	✓	✗	✗
[7]	Free5GC	✓	✓	✗	✗
[6]	OAI EPC	✗	✓	✗	✗
[10]	Open5GS	✓	✓	✗	✗
[3]	OAI 5GC	✓	✓	✗	✗
[11]	Open5GS + OAI 5GC	✓	✓	✗	✗
Our work	OAI 5GC	✓	✓	✓	✓

broader access to core-network data. A follow-up study in [9] extended this analysis within the same environment. However, both works rely on passive packet capture rather than standardized event-exposure services, limiting real-time interaction with NFs and preventing closed-loop automation. In [3], the authors presented a microservice-based NWDAF aligned with 3GPP guidelines, aggregating data from core NFs, virtualized infrastructure managers, and xApps. Their system supports core analytics and ML-driven intelligence, including an LSTM autoencoder for traffic anomaly detection. Nonetheless, the design employs a statically deployed ML model and lacks automated or continuous model training, which is essential in 3GPP Release 17 for MTLF and its interaction with the AnLF.

Peters *et al.* [10] developed an NWDAF tailored for anomaly detection in private 5G networks by extending Open5GS to provide limited event-exposure data from the Access and Mobility Management Function (AMF) and the Session Management Function (SMF). Although effective for identifying operational anomalies, the system does not implement automated mitigation or any training capabilities. A more recent effort [11] introduced an NWDAF integrated with both OAI and Open5GS, incorporating the UPF Event Exposure Service for standardized real-time data collection. While the design supports ML model provisioning and dual-core interoperability, it lacks automated model training, requiring offline manual model management for each analytics use case and constraining system adaptability.

As summarized in Table I, existing open-source NWDAF initiatives, despite their valuable contributions, continue to exhibit several significant limitations: (i) none comply with 3GPP Release 17 by implementing the AnLF–MTLF separation, (ii) none provide a standards-aligned ML model-training pipeline, leaving model lifecycle management either incomplete or entirely absent, and (iii) all lack essential MLOps capabilities such as automated training, versioning, continuous delivery, and performance monitoring. To address these limitations, this paper presents a new service-based NWDAF implementation that fully supports the Release 17 architecture, standardized event-exposure services, and a complete MLOps-driven model lifecycle, paving the way toward autonomous 6G networks.

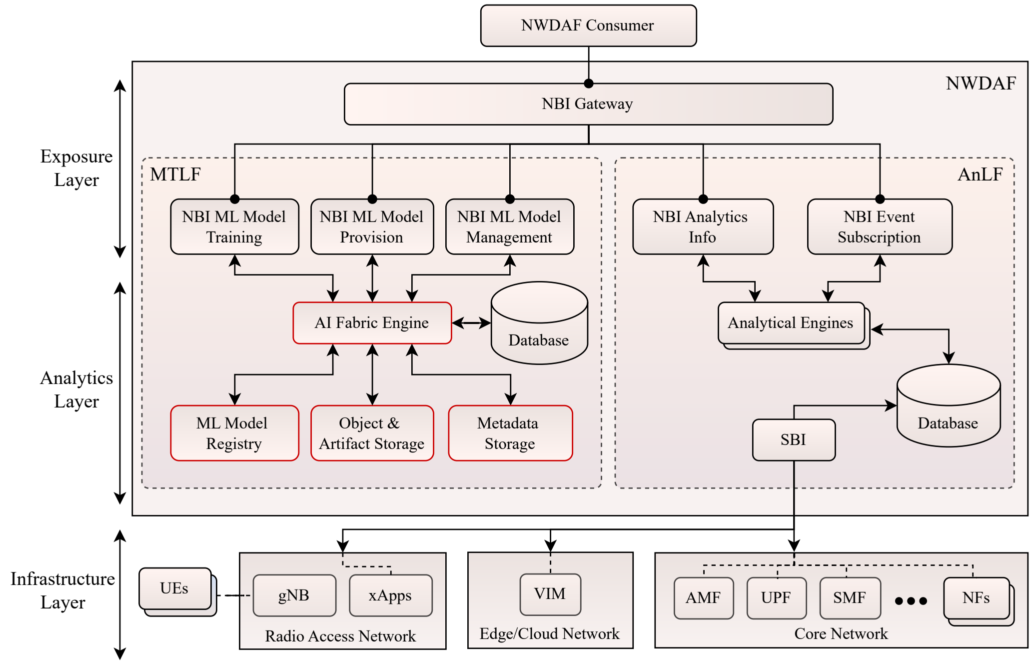


Fig. 1: The proposed service-based NWDAF Release 17 architecture. Red-highlighted components indicate the MLOps services within the MTLF.

III. NWDAF Design

This section presents the proposed NWDAF architecture, highlighting the layered AnLF–MTLF components.

A. *Overall System Architecture*

The proposed NWDAF architecture adopts a structured multi-layer design, as illustrated in Fig. 1. It consists of (i) the *Exposure Layer*, (ii) the *Analytics Layer*, and (iii) the *Infrastructure Layer*. Together, these layers support telemetry acquisition, analytics generation, MLOps workflows, and standards-compliant service exposure. The Exposure Layer provides Northbound Interfaces (NBI) for submitting analytics requests and event subscriptions. The Analytics Layer processes these requests by gathering data from the Infrastructure Layer and executing statistical or MLOps-driven tasks. The Infrastructure Layer supplies data from the Radio Access Network (RAN), edge/cloud platforms, and core NFs. Through this structure, the architecture delivers a service-based design that supports both inference and model training in alignment with 3GPP Release 17.

B. *Exposure Layer*

The Exposure Layer serves as the system entry point, managing all interactions between NWDAF and its consumers, including NFs and third-party services. As shown in Fig. 1, its central element is the NBI Gateway, which provides a standards-compliant interface and routes incoming requests to the appropriate component. For the AnLF, two service categories are exposed. The NBI Analytics Info module handles analytics queries by forwarding requests to the Analytics Layer and assembling responses, enabling consumers to access real-time or on-demand network insights. The NBI Event Subscription module manages event subscriptions, evaluates conditions via the Analytics Layer, and delivers notifications when events are triggered. Together, these modules provide flexible, standards-aligned exposure for both pull-based analytics and push-based notifications. For the MTLF, three MLOps-oriented services are exposed. The ML Model Training NBI and ML Model Provision NBI provide standardized APIs for model training and inference, respectively, while the ML Model Management NBI offers operators full lifecycle management of ML models (e.g., model retraining, versioning, monitoring, etc.).

C. *Analytics Layer*

The Analytics Layer constitutes the core computational engine of NWDAF. Each network analytics service corresponds to an engine deployed within the AnLF and MTLF. The Exposure Layer selects the appropriate engine according to the requested service type. Analytical Engines may rely solely on statistical computations from AnLF (e.g., number of attached UEs) or may require predictive output, in which case coordination with the MTLF becomes necessary (e.g., abnormal behavior detection). When an analytics engine requires ML inference, the AnLF queries the MTLF through the ML Model Provision service to determine whether an appropriate model exists. The MTLF leverages its Model Registry to check for an existing model. If a suitable model is available, the MTLF deploys the model through its internal AI Fabric Engine and provides the AnLF with an inference endpoint.

When a required model is missing or outdated, the MTLF initiates a model training workflow using the ML Model Training service. The training pipeline retrieves datasets from the object storage (if the dataset is available) or directly from the Infrastructure Layer, executes the training procedure, stores the resulting artifacts in the Object and Artifact Storage, and records model metadata such as version information, accuracy, and training parameters in the Metadata Storage. Once training completes, the MTLF notifies the AnLF that the model is ready for provisioning. This dual workflow enables NWDAF to support both inference and training pipelines in full compliance with 3GPP service exposure specifications, with the MTLF's MLOps services providing dynamic, self-contained ML lifecycle management within the analytics process.

D. Infrastructure Layer

The Infrastructure Layer comprises the underlying data sources spanning the RAN, edge/cloud platforms, and the core network. The Southbound Interface (SBI) orchestrates all data collection procedures and manages subscriptions to NFs to enable continuous monitoring, as depicted in Fig. 1. At initialization, the SBI establishes subscriptions to core NFs to obtain status updates and performance notifications, which are then stored in the system database. The SBI also requests resource-level telemetry, such as CPU and memory utilization, from the Virtual Infrastructure Manager (VIM). In parallel, NWDAF collects RAN statistics through O-RAN mechanisms using xApps, following O-RAN Alliance specifications. These heterogeneous data streams collectively populate the analytics pipeline and provide comprehensive visibility across the network.

Overall, the proposed NWDAF offers a modular and scalable framework that integrates standards-compliant service exposure, distributed analytics computation, and MLOps-enabled intelligence closely coupled with diverse network data sources.

IV. Performance Evaluation

This section describes the NWDAF implementation details, outlines the experimental setup, and presents the performance evaluation results.

A. Implementation Details

The proposed NWDAF is implemented using a heterogeneous open-source stack optimized for performance, scalability, and 3GPP standards compliance. Notably, we extend the OAI NWDAF[2] baseline to implement the AnLF components. These components are developed in Golang for high-performance execution. Communication is managed through a flexible, scalable RESTful API in the Exposure Layer. For persistent storage of semi-structured and unstructured network data, MongoDB serves as the primary data store. For the MTLF, the ML Model Training and ML Model Provision NBIs are implemented in Golang, while Python and FastAPI are used for model-management exposure and the AI Fabric. MLflow[3] serves as the model registry, MinIO[4] handles datasets and artifacts, and PostgreSQL stores model metadata, including performance metrics and version history. Together, these components form a service-based MLOps framework supporting both training and inference pipelines while meeting NWDAF exposure requirements. The network environment uses the OAI Core Network[5] to emulate 5G core functions (AMF, SMF), gnbsim to simulate RAN events, and Iperf to generate realistic UE traffic. This setup provides a controllable, reproducible platform for validating NWDAF analytics and MLOps workflows.

B. Experimental Setup

We validate the NWDAF implementation through experiments evaluating both MTLF and AnLF capabilities using standardized 3GPP use cases [4]. For the MTLF, two representative scenarios were implemented: abnormal behavior detection and network performance forecasting, corresponding to the NWDAF event identifiers ABNORMAL_BEHAVIOR and NETWORK_PERFORMANCE. The abnormal behavior model is an Autoencoder-LSTM implemented in TensorFlow with a two-layer encoder (32 and 16 LSTM units) and symmetric decoder, processing sequences of length 12 standardized via StandardScaler. The network was optimized using Adam to minimize mean squared error, trained through the service-based MTLF, registered in MLflow, and stored in MinIO with metadata in PostgreSQL. Whereas the NETWORK_PERFORMANCE model is a stacked LSTM with two recurrent layers (128 and 64 units), a 24-step lookback window, 0.2 dropout, and a dense output neuron for regression. Training used Adam with a 0.001 learning rate, batch size 32, and early stopping with patience 10. For the AnLF, multiple analytics services were deployed. UE attach attempts were computed from AMF registration notifications, session success rate from SMF PDU (Packet Data Unit) session establishment notifications, and UE communication statistics from SMF usage reports collected via the UPF N4 interface. NFs load was derived from CPU and RAM metrics provided by the VIM. All experiments ran on a workstation with a 10-core processor operating at 5.20 GHz and 32 GB RAM.

C. Experimental Results

In this section, we present a detailed evaluation of the proposed NWDAF, examining both the AnLF and MTLF components. The analysis places particular emphasis on the MTLF training and inference pipelines, which represent the core contribution of this work.

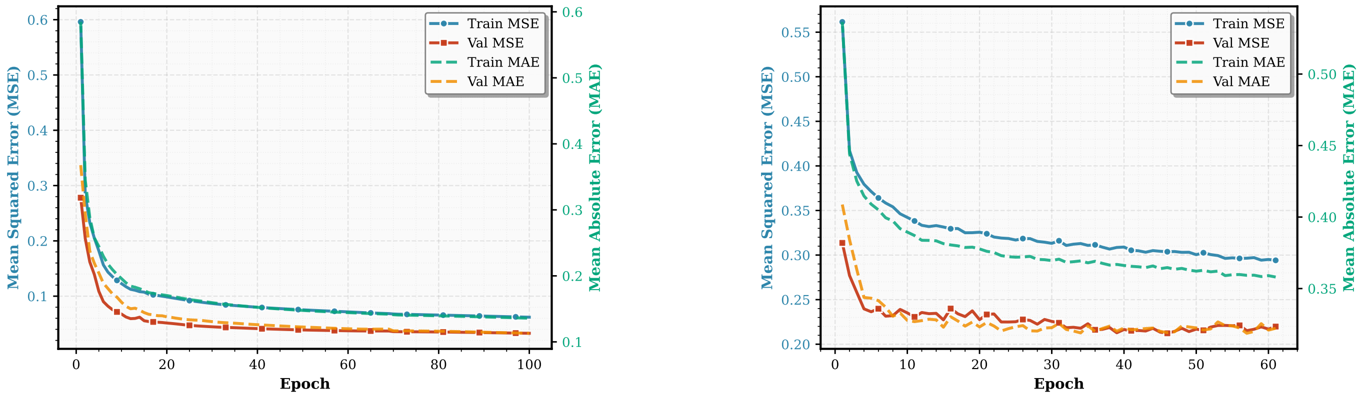
1) MTLF Training Validation: To evaluate the effectiveness of the MTLF training pipeline, Fig. 2 presents the training and validation curves for both implemented use cases: abnormal behavior detection and time series traffic forecasting. As depicted in Fig. 2a, the abnormal behavior model exhibits a

[2]https://gitlab.eurecom.fr/oai/cn5g/oai-cn5g-nwdaf

[3]https://mlflow.org/

[4]https://www.min.io/

[5]https://gitlab.eurecom.fr/oai/cn5g



(a) Abnormal Behavior LSTM-Autoencoder Model Training within NWDAF.

(b) Time Series Traffic Forecasting LSTM Model Training within NWDAF.

Fig. 2: NWDAF-MTLF Training Capabilities.

TABLE II: NWDAF-MTLF Model Versioning Capabilities.

Use Case	Model Version	MSE	MAE	R^2
Abnormal behavior	v1.0	0.0280	0.1300	0.8300
Traffic forecasting	v1.0	0.0210	0.1067	0.8442
Abnormal behavior	v2.0	0.0180	0.0950	0.8820
Traffic forecasting	v2.0	0.0163	0.0824	0.8922
Abnormal behavior	**v3.0**	**0.0140**	**0.0750**	**0.9320**
Traffic forecasting	v3.0	0.0122	0.0670	0.9400
Abnormal behavior	v4.0	0.0143	0.0760	0.9300
Traffic forecasting	v4.0	0.0121	0.0660	0.9420
Abnormal behavior	v5.0	0.0148	0.0780	0.9280
Traffic forecasting	**v5.0**	**0.0116**	**0.0650**	**0.9436**

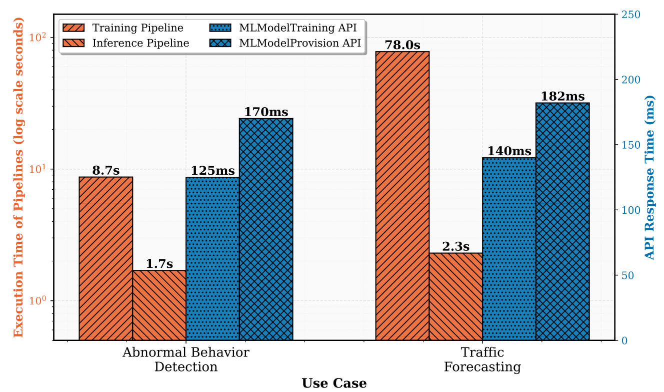


Fig. 3: Execution Time Performance of Training/Inference Pipelines and API Exposure.

rapid decline in both training and validation MSE and MAE within the first few epochs, followed by smooth convergence, indicating stable reconstruction learning and absence of overfitting. Similarly, the forecasting model (see Fig. 2b) demonstrates consistent reductions in error metrics, with validation curves closely tracking the corresponding training curves. This behavior confirms that the MTLF orchestrates the training workflow correctly, manages datasets and artifacts as intended, and produces models with reliable generalization performance.

2) Model Versioning Validation: Model versioning is essential for maintaining reliable and reproducible analytics within NWDAF. Table II summarizes the performance of successive model versions generated through repeated training cycles within the MTLF. The results demonstrate clear performance improvements, with version v3.0 achieving the highest overall accuracy for the abnormal behavior use case, and version v5.0 providing the best accuracy for the traffic forecasting task. This validates the effectiveness of the integrated MLflow registry in managing model evolution. Model versioning is essential for supporting hyperparameter optimization, allowing multiple candidate models to be tracked and compared, while also enabling safe rollback to a previous version when newer models underperform. These capabilities ensure that NWDAF consistently provisions the most reliable analytics models in accordance with 3GPP-aligned MLOps workflows.

3) MTLF Closed-loop Execution Time Performance: In this experiment, we evaluate the execution time performance of the MTLF training and inference pipelines, along with the response times of the MLModelTraining and MLModelProvision service exposure APIs, as shown in Fig. 3. The two use cases exhibit different computational demands. The Autoencoder for abnormal behavior detection completes training in 8.7 s, whereas the stacked LSTM for traffic forecasting requires 78.0 s due to its larger hidden layers and longer temporal window. Inference times follow a similar trend, with the Autoencoder completing inference in 1.7 s and the forecasting model in 2.3 s, both within the bounds required for near real-time NWDAF analytics. API response times are consistently low for both workflows. MLModelTraining requests complete in 125–140 ms, and MLModelProvision calls in 170–182 ms. Latency is dominated by request processing rather than model computation, indicating that the exposure layer introduces minimal overhead and supports timely interaction between the AnLF and MTLF. It is worth noting that the end-to-end pipeline execution time includes the API response time. Overall, these results demonstrate that the MTLF scales effi-

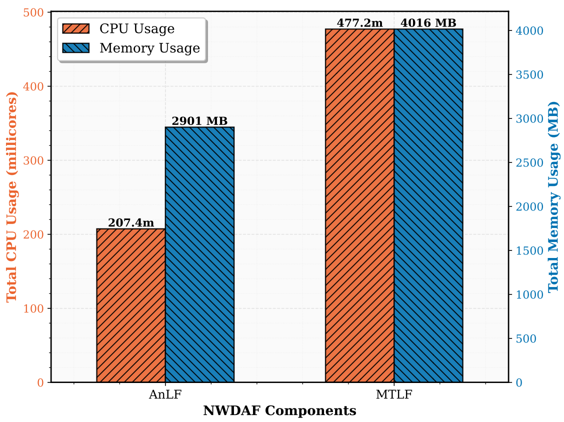


Fig. 4: Resource Usage Comparison between AnLF vs MTLF.

ciently with model complexity while maintaining low-latency inference and fast API responsiveness.

4) AnLF-MTLF Resource Usage: To study the resource utilization characteristics of NWDAF, we evaluated and compared the CPU and memory consumption of the AnLF and MTLF components, as shown in Fig. 4. The AnLF exhibits relatively low resource usage, with 207.4 millicores of CPU and 2901 MB of memory. This behavior is consistent with its role as a lightweight inference and analytics engine, where computations are primarily statistical or involve invoking pre-provisioned ML models. By contrast, the MTLF requires significantly higher resources, consuming 477.2 millicores of CPU and 4016 MB of memory. This increased load stems from executing training pipelines, handling datasets, managing MLflow services, and coordinating artifact and metadata storage. These operations are inherently more computationally intensive and memory demanding than inference. This validates the practicality of integrating a service-based MLOps framework directly into NWDAF.

D. Discussion & Learned Lessons

The comprehensive evaluation of the proposed NWDAF yields several important insights and lessons learned. First, integrating end-to-end MLOps capabilities into the MTLF is both technically feasible and operationally advantageous. Furthermore, the architectural separation between AnLF and MTLF emerges as a critical design choice, enabling real-time inference while offloading computationally intensive training tasks to dedicated components. In addition, the experiments confirm that automated model lifecycle management is essential in dynamic mobile networks, where evolving traffic patterns necessitate continuous retraining, controlled versioning, and reliable rollback. Likewise, the service-based interaction between analytics and training components introduces minimal overhead, thereby preserving inference performance consistent with 5G operational requirements. Moreover, resource usage results emphasize the need for proper provisioning, since training workloads can impose notable CPU and memory demands. Collectively, these findings validate the necessity of an MLOps-enabled MTLF for meeting 3GPP Release 17 expectations and provide practical guidance for scaling NWDAF capabilities as networks advance toward autonomous, data-driven operation in future 5G/6G systems.

V. Conclusion

In this paper, we presented a new NWDAF implementation that is fully compliant with 3GPP Release 17 and supports autonomous network operation. Building on the OAI core network, we introduced a redesigned NWDAF architecture that adheres to the AnLF–MTLF logical split and integrates a service-based MLOps platform exposing standardized APIs for model training and provisioning. This architecture advances beyond existing open source initiatives by enabling seamless ML workflow orchestration and supporting automated model training, deployment, and inference, thereby demonstrating closed-loop intelligence across multiple 3GPP-defined analytics use cases. **Future work** will extend this NWDAF release with advanced Federated Learning capabilities aligned with the 3GPP Release 18 vision.

Acknowledgment

This work was supported by the European Union's Horizon Program under 6G-DALI (Grant Agreement No. 101192750) and SUNRISE-6G projects (Grant No. 101139257).

References

[1] T. A. Kuzovkova, O. I. Sharavova and M. M. Sharavova, "The Role of 5G and 6G Mobile Communication Technologies for Industry 4.0 and the Transition to Industry 5.0," 2024 International Conference on Engineering Management of Communication and Technology (EMCTECH), Vienna, Austria, 2024, pp. 1-6, doi: 10.1109/EMCTECH63049.2024.10741817.

[2] M. Ameur, B. Brik and A. Ksentini, "Dual Self-Attention is What You Need for Model Drift Detection in 6G Networks," in IEEE Transactions on Machine Learning in Communications and Networking, vol. 3, pp. 690-709, 2025, doi: 10.1109/TMLCN.2025.3576727.

[3] A. Mekrache, K. Boutiba and A. Ksentini, "Combining Network Data Analytics Function and Machine Learning for Abnormal Traffic Detection in Beyond 5G," GLOBECOM 2023 - Kuala Lumpur, Malaysia, 2023, pp. 1204-1209, doi: 10.1109/GLOBECOM54140.2023.10436766.

[4] 3GPP, 5G System; Network Data Analytics Services; Stage 3, 3rd Generation Partnership Project (3GPP), Technical Specification 29.520, V18.6.0, Rel. 18, July 2024.

[5] F. Rezazadeh, H. Chergui, L. Alonso, and C. Verikoukis, "SliceOps: Explainable MLOps for streamlined automation-native 6G networks," IEEE Wirel. Commun., vol. 31, no. 5, pp. 224–230, 2024.

[6] K. Abbas, T. A. Khan, M. Afaq, J. J. Diaz Rivera, and W.-C. Song, "Network data analytics function for IBN-based network slice lifecycle management," in APNOMS, 2021.

[7] A. A. Bayleyegn, Z. Fernández, and F. Granelli, "Real-time monitoring of 5G networks: An NWDAF and ML based KPI prediction," in 2024 IEEE 10th International Conference on Network Softwarization (NetSoft), 2024.

[8] A. Chouman, D. M. Manias, and A. Shami, "Towards supporting intelligence in 5G/6G core networks: NWDAF implementation and initial analysis," in IWCMC, 2022.

[9] D. M. Manias, A. Chouman, and A. Shami, "An NWDAF approach to 5G core network signaling traffic: Analysis and characterization," in GLOBECOM 2022 - IEEE Global Communications Conference, 2022.

[10] S. Peters, F. Sivrikaya, and A. Satyatma Winarga Rochadi, "Cobra-5G - an AI-Driven Solution for Resilient Industrial Applications in Private 5G Environments," in ICCNS, 2024, pp. 92–99.

[11] F. S. Ardestani, N. Saha, N. Limam, and R. Boutaba, "Towards NWDAF-enabled analytics and closed-loop automation in 5G networks," arXiv [cs.NI], 2025.