A Protocol-Based Framework for AIaaS Lifecycle Management in 6G via NWDAF

Júnia Maísa Oliveira*, **Daniel Fernandes Macedo**† and **José Marcos Nogueira**†**,** *Member, IEEE*
* Department of Electrical, Electronic, and Information Engineering – University of Bologna, Italy
†Computer Science Department - Universidade Federal de Minas Gerais, Brazil
E-mail: junia.deoliveira@unibo.it, {damacedo, jmarcos}@dcc.ufmg.br

***Abstract*—6th-generation (6G) mobile networks are envisioned as AI-native systems, integrating learning and inference across the entire protocol stack. Although 5G's 3GPP Network Data Analytics Function (NWDAF) introduced analytics-driven automation, it lacks standardised support for model lifecycle control, Data Analytics as a Service (DAaaS), closed-loop feedback, and large-scale interoperability. To address these gaps, we propose a protocol-based framework for AI-as-a-Service (AIaaS) management for 6G, centered on an enhanced NWDAF architecture with four components: Model Lifecycle Orchestrator, Model Registry & Validator, Distributed Execution Engine, and Feedback Aggregator. It introduces two lightweight, service-based interfaces: *Model Training and Creation Protocol (MTCP)* for intent-based model training and publication, and *Model Execution Protocol (MEP)* for on-box inference and metric feedback. We validate the framework via formal verification under message loss using a reproducible TLA+ model with three NFs and two model versions. Results show that NWDAF can evolve into a feasible AI lifecycle manager, enabling scalable and stable AI-native deployments in 6G. Complexity modelling confirms linear resource scaling up to 128 network functions (theoretical), with the public TLA+ specification configured for 3 NFs.**

***Index Terms*—AI-as-a-Service, 6G, NWDAF, Data analysis, Artificial Intelligence, 3GPP.**

I. Introduction

Sixth-Generation (6G) standardization is transitioning from vision-setting to formal studies and early system blueprints. ITU-R approved the IMT-2030 framework (Rec. M.2160) in 2023, defining capabilities and usage scenarios for "6G" [1], while Third Generation Partnership Project (3GPP) scheduled 6G technical studies in Release 20 starting June 2025 and identified Release 21 as the first phase for normative 6G specifications [2]. 6G mobile networks are anticipated to be fundamentally *AI-native*, with integration of machine learning, inference, and optimisation capabilities throughout the network architecture [3], [4]. Although the 3GPP introduced the Network Data Analytics Function (NWDAF) in 5G Release 16 to bootstrap analytics-driven automation, its current implementations remain largely *centralised* and *model-agnostic* that lacking native support for AI/ML lifecycle management. Recent studies have proposed enhanced frameworks, e.g., INTDAI [5], Hierarchical NDAF (H-NDAF) [6], and e-NWDAF [7] – that distribute intelligence closer to network functions (NFs); however, the community still lacks a *standardised lifecycle protocol* that orchestrates data collection—often provisioned via data analytics provisioning datasets (DAaaS) components—training, validation, deployment, and continuous improvement of machine-learning (ML) models.

Federated and split learning paradigms [8], [9] promise scalability and privacy preservation, yet their integration with core and edge domains demands a model manager capable of versioning, policy enforcement, and feedback integration. In parallel, 3GPP TR 23.288 and TS 29.520 highlight open issues in *model activation, monitoring, and fallback* that hinder operational adoption. These requirements underscore the critical need for a standardized AI-as-a-Service (AIaaS) framework that transforms the NWDAF from a passive analytics aggregator into an intelligent orchestration platform for distributed AI/ML workflows across 6G networks.

This work investigates the feasibility of transforming the NWDAF from a passive analytics component into a comprehensive AI lifecycle manager for 6G networks. We propose this using standard-compliant, lightweight protocols that cover training, deployment, and feedback across all network functions (NFs). Our research makes four key contributions: A **modular, 6G-ready NWDAF architecture** that decouples orchestration, execution and feedback planes while remaining fully aligned with 3GPP Service-Based Architecture (SBA); A **Model Training and Creation Protocol (MTCP)** that formalises intent-based requests, data collection, training and versioned publication of models; A **Model Execution Protocol (MEP)** that enables secure model retrieval, on-box inference and closed-loop metric reporting within network functions; A **theoretical evaluation** comprising (i) formal verification of protocol correctness, (ii) control-theoretic proof of loop stability under 3GPP latency budgets, and (iii) analytical scalability modelling of CPU and storage costs.

Unlike prior work centred on isolated use cases (e.g., UPF migration [5] or throughput prediction [6]), our framework is *task-agnostic* and intended as a reusable

978-3-903176-75-1 ©2025 IFIP

template for AIaaS deployment across diverse network domains.

II. Related Work

Multiple research efforts have recently advanced the vision of AIaaS in 5G/6G networks. Below, we group them by key themes and outline the remaining gaps that our work addresses.

A. AI–Native Frameworks and NWDAF Evolution

Majumdar *et al.* [5] embed intelligent agents directly in network functions through INTDAI, enabling distributed training and inference with measurable latency gains. Jeon and Pack [6] separate a central training root from distributed leaves in H–NDAF, demonstrating throughput prediction accuracy in free5GC. Moreira *et al.* [7] present the evolved NWDAF (e–NWDAF) with intent-based services, while Nadar and Härri [10] introduce microservice-oriented model provisioning and semantic matchmaking. These works confirm the relevance of decentralised analytics yet omit a *standardised protocol suite* to manage the entire lifecycle of models.

B. Federated and Distributed Learning for AIaaS

Lu *et al.* [8] review federated learning (FL) as a privacy-preserving enabler of AIaaS, whereas Li *et al.* [11] extend FL to user equipment via the *Personal AI* concept. Li *et al.* [9] advocate a layered AI-native architecture supporting FL, Split, and Swarm Learning paradigms. These studies stress the importance of distributed training but do not define *interoperable mechanisms* for versioning, validation, or feedback once models are deployed across heterogeneous network domains.

C. Standardisation Status and Architectural Insights

Sun *et al.* [4], Yeh *et al.* [12] and Lin *et al.* [13] survey 3GPP progress on NWDAF, RAN Intelligent Controller (RIC) and AI in wireless. Matera *et al.* [14] highlight orchestration gaps, while Liu *et al.* [3] and Yi *et al.* [15] argue for native intelligence and secure AI provisioning. Although these works identify challenges in model management, monitoring, and fallback, they stop short of prescribing concrete message flows or control loops.

D. Large–Scale AI Services and LLM Integration

Tarkoma *et al.* [16] envision *AI Interconnect*, integrating large language models (LLMs) via MAPE–K patterns, emphasising orchestration but lacking detailed NWDAF interaction schemes.

Table I: Comparison with existing AI–analytics frameworks

Framework	Lifecycle protocol	Feedback loop	Versioned catalogue
INTDAI [5]	✗	✓	✗
H-NDAF [6]	✗	✓	✗
e-NWDAF [7]	✗	✗	✗
AI Interconnect [16]	✗	✗	✗
This work	✓	✓	✓

✓ feature addressed; ✗ not addressed.

E. Identified Gaps

Table I summarises how representative frameworks cover (or overlook) gaps below identified, confirming that no prior proposal addresses all three simultaneously. Across the above literature, we observe three persistent limitations: **Lifecycle Protocol Absence**: No end-to-end protocol set exists for request, training, validation, publication, consumption and retirement of AI models; **Feedback and Drift Handling**: Works rarely address closed-loop reporting from inference back to training entities for automated re-training triggers; **Interoperability at Scale**: Semantic discovery and versioned model catalogues are discussed conceptually but lack formal interface definitions aligned with 3GPP SBA.

This paper targets the above gaps by specifying two lightweight, SBA-compliant protocols:
(i) the ***Model Training and Creation Protocol (MTCP)*** orchestrates data collection, federated or centralised training, validation, and publication;
(ii) the ***Model Execution Protocol (MEP)*** enables secure model retrieval, on-box inference, and metric feedback. Together, they operationalise NWDAF as an *AI lifecycle manager*, complementing and extending prior frameworks.

III. Proposed Architecture

Fig. 1 depicts the proposed **NWDAF-AIaaS Stack**. It comprises four logical components, mapped onto a 3GPP SBA via Network Function (NF) and Network Exposure Function (NEF):

1) **Model Lifecycle Orchestrator (MLO)**—extends NWDAF control capabilities to handle NFs*intent-based* model requests, negotiate data schemas, and schedule training tasks across cloud/edge resources.
2) **Model Registry and Validator (MRV)**—maintains a versioned catalogue of models with cryptographic signatures, validation scores, and domain-specific metadata to support audit and rollback.
3) **Distributed Execution Engine (DEE)**—a lightweight runtime embedded within each NF (e.g., AMF, SMF, UPF) capable of loading ONNX /TensorRT artefacts, executing inference under tight latency budgets, and exporting performance counters.
4) **Feedback Aggregator (FA)**—collects inference outcomes, drift indicators, and resource metrics,

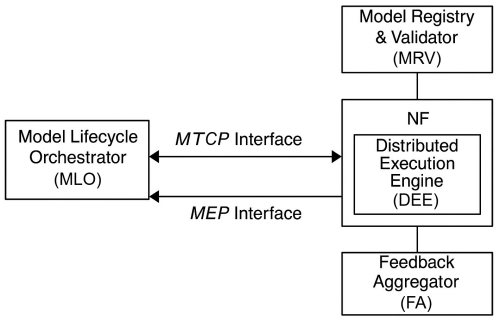

Figure 1: NWDAF-AIaaS architecture with orchestration, execution, and feedback planes.

triggering retraining workflows when SLAs degrade.

O stack builds upon 3GPP service-based interfaces and assumes a DAaaS-compatible backend for data provisioning. It introduces two new APIs—`/mtcp` and `/mep`—whose message sequences are detailed in Section IV.

IV. Lifecycle Protocols

This section formalises the two proposed protocols that operationalise the model lifecycle, MTCP and MEP. MTCP/MEP are proposed service interfaces aligned with SBA; they are not 3GPP specifications.

A. Model Training and Creation Protocol (MTCP)

MTCP governs the end-to-end creation of models through the **five phases** illustrated in Fig. 2. It consists of five phases and leverages JSON/HTTP-2 messages secured via OAuth 2.0 tokens, scopes compliant with 3GPP TS 29.510 [17].

1) **MTCP–Request**: The NF submits an *Intent* specifying the analytic task (e.g., congestion prediction), latency/SLA constraints, and data-source URIs.
2) **MTCP–DataCollect**: The MLO instructs Data Exposure Services (via NEF) to stream schema-aligned training data to the designated compute domain (edge or cloud).
3) **MTCP–Train**: Training is executed according to the chosen paradigm (centralised, federated, or split), producing candidate checkpoints.
4) **MTCP–Validate**: The Model Registry & Validator (MRV) performs offline accuracy tests, bias checking, and cryptographic signing; only models meeting policy thresholds advance.
5) **MTCP–Publish**: The validated artefact receives an immutable `modelURI`; previous versions are marked `retired` and the new model becomes discoverable via the `/mep` interface.

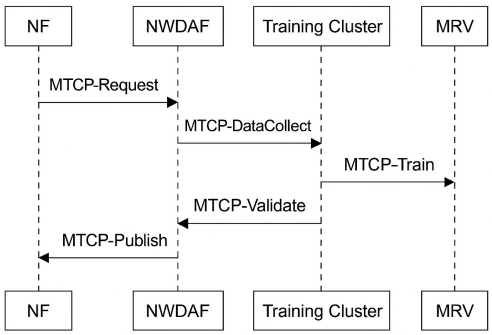

Figure 2: Message sequence for MTCP lifecycle (five phases).

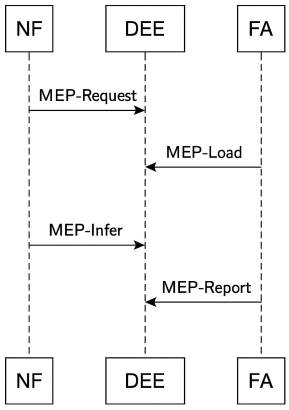

Figure 3: Message flow for MEP runtime operation.

B. Model Execution Protocol (MEP)

MEP enables NFs to consume and monitor models during operation (Fig. 3). It provides *on-demand* and *subscription* modes to balance latency and bandwidth.

1) **MEP-Request**: The NF requests a `modelURI` with desired execution profile (batch/stream), expected inference rate, and QoS hints.
2) **MEP-Load**: DEE resolves dependencies and instantiates the model; success or fallback is acknowledged to the MLO.
3) **MEP-Infer**: Real-time inference occurs within NF process space, minimising cross-domain latency.
4) **MEP-Report**: DEE streams inference KPIs (e.g., accuracy, resource usage) and drift statistics to the FA; threshold violations trigger an MTCP-Request for retraining.

V. Theoretical Evaluation

This section validates the feasibility of the proposed NWDAF–AIaaS framework through *three complementary theoretical methods*: 1) formal verification [18], 2) control-theoretic stability analysis [19] and 3) scalability/complexity modelling analysis. The goal is to provide deployment-level guarantees without requiring large testbeds. To support review reproducibility and

artifact availability, we provide a public repository for the formal model and configuration [20].

A. Formal Verification of Lifecycle Protocols

We specify the **MTCP** and the **MEP** as communicating extended finite–state machines in TLA$^+$ and verify them with the TLC explicit–state model checker, Lamport's methodology [18]. TLA$^+$ captures state variables, actions, and temporal properties in one model; TLC explores finite instantiations and returns a reproducible report (states explored, search depth, satisfied properties).

Each participating entity (NF, MLO, MRV, DEE, and FA) is modeled as an independent process with typed variables (e.g., `modelURI`, `state {draft,valid,published,retired}`) and authorization guards aligned with OAuth 2.0 scopes per 3GPP TS 29.510 [17]. In this context, *safety* (S) ensures nothing goes wrong, while *liveness* (L) guarantees that desired outcomes eventually occur [21]. We formally verify the following five primary properties:

- **S1 — Exactly-once publication:** a given `modelURI` is published at most once throughout execution.
- **S2 — No stale load:** models previously marked as `retired` cannot be loaded or reused by any NF.
- **S3 — Deadlock freedom:** the global system never reaches a deadlock; the system always maintains at least one enabled transition, preventing total halt.
- **L1 — Eventual publication:** every MTCP-Request eventually leads to an MTCP-Deploy. Every MTCP intent request by an NF eventually leads to a model being published.
- **L2 — Eventual feedback:** each MEP-Infer is followed, within a bounded number of transitions, by an MEP-Report. Each inference executed under MEP is followed, within a bounded delay, by a feedback report that may trigger retraining.

In total, **eight temporal properties** were verified: the five main properties (three safety and two liveness), and three auxiliary properties—*NoDuplicateModelURI*, *ProgressForEachNF*, and *AlwaysEventuallyValidState*—which ensure uniqueness, individual NF progress, and overall lifecycle consistency.

Using TLC, the reduced model configured with **three network functions and two model versions explored 18,050 states (65 distinct), with a maximum search depth of 11**. No invariant violations or deadlocks were reported. All properties were satisfied, confirming that MTCP/MEP ensures correctness even under message loss and reordering. The full specification and TLC configuration are publicly available at the AIasServ-LifeCycle repository [20].

B. Control-Theoretic Stability of the Closed Loop

In a UPF load-balancing scenario, the Distributed Execution Engine (DEE) runs the selected model inside the UPF process. At each sampling instant k with period T_s, the UPF exports the KPI $x[k]$ (e.g., per-tunnel buffer occupancy), while the on-box inference returns the prediction $\hat{x}[k]$ via MEP-INFER. The prediction error $e[k] = x[k] - \hat{x}[k]$ is streamed by MEP-REPORT to the Feedback Aggregator (FA). When thresholds are exceeded, the FA issues an MTCP-REQUEST that triggers (re)training and publication by the Model Lifecycle Orchestrator (MLO). The end-to-end actuation delay is d samples.

Linearizing the closed loop yields the discrete-time transfer function

$$G(z) = \frac{K\, z^{-d}}{1 - a\, z^{-1}}, \tag{1}$$

where K is the adaptation gain and $|a| < 1$ is the residual error factor. By the Jury stability criterion, the loop remains stable if

$$|K| < 1 - |a| \quad \text{and} \quad d\,T_s < \tau_{\max}, \tag{2}$$

with $\tau_{\max}$ denoting the staleness bound mandated by 3GPP TR 23.288 for edge applications.

For $T_s = 100\,\text{ms}$, $a = 0.25$, and $\tau_{\max} = 500\,\text{ms}$, we obtain

$$K_{\text{crit}} = 1 - |a| = 0.75, \qquad d \le \frac{\tau_{\max}}{T_s} = 5,$$

which defines the admissible adaptation gain and maximum actuation delay (in samples) for the UPF controller.

C. Scalability and Complexity Analysis

Let N be the number of NFs (e.g., UPF instances), V the number of model versions, S the average model size, f_{inf} the inference rate per NF, and c_{rpc} the CPU cost per RPC. For the UPF use case, one RPC is issued per inference to NWDAF/DEE control:

$$\text{CPU}_{\text{NWDAF}} = N\, f_{\text{inf}}\, c_{\text{rpc}}, \qquad \text{Storage}_{\text{MRV}} = N\, V\, S.$$

Adopted parameters:

- $c_{\text{rpc}} = 12\,\mu\text{s}$ — one-way latency from the Istio v1.18 benchmark [22].
- $f_{\text{inf}} = 1\,\text{kHz}$ — expected control-plane trigger rate (3GPP TR 23.288) [23].
- $S = 12\,\text{MiB}$ — size of a compressed MobileNet-v2 from the ONNX Model Zoo [24].

Assuming $N = 128$ UPFs for theoretical extrapolation, the NWDAF load is $\approx N f_{\text{inf}} c_{\text{rpc}} = 128 \times 10^3 \times 12\,\mu\text{s} \approx 1.6$ CPU cores ($< 10\%$ of a 16-core edge node). With $V = 5$ and $S = 12\,\text{MiB}$, total storage is $NVS = 7.5\,\text{GiB}$. Both CPU and storage scale linearly with N (and storage with V). **Note:** The public TLA+ model uses $N = 3$, $V = 2$ for verification tractability.

VI. Conclusion

This work proposes to transform the 3GPP-NWDAF from a passive metrics collector to a complete *AI-lifecycle manager* for 6G networks. We introduced a modular architecture with four logical planes (MLO, MRV, DEE, FA) and specified two service-based interfaces, MTCP and MEP, that together cover intent-based model requests, data collection, training, validation, publication, on-box inference, and closed-loop feedback. A theoretical evaluation proved that the protocols are (i) functionally correct under exhaustive model checking, (ii) control-loop stable within 3GPP latency budgets, and (iii) linearly scalable with modest CPU and storage overheads. All results derive from formal reasoning and analytical formulas. The analysis assumes a trusted NF–NWDAF channel and does not model adversarial threats such as message tampering or model-poisoning attacks. Energy consumption of distributed training jobs also remains outside the current scope. The Theoretical Evaluation is presented in GitHub repository [20]. In future work, we will consider model re-training and fine-tuning. Next steps will operationalize MTCP/MEP in the published scalable NWDAF-based framework of De Oliveira *et al.* [25] over Free5GC, adding /mtcp and /mep endpoints, OAuth 2.0 scope enforcement, and conformance tests, then repeating the framework's experiments to quantify latency, per-NF inference throughput, and storage.

ACKNOWLEDGMENT

This work has been partially funded by the Brazilian institutions, CAPES, CNPq, FAPEMIG, and FAPESP/MCTI (grants 2023/13518-0, 2020/05182-3, and 2018/23097-3). The views expressed are those of the authors and do not necessarily represent the supporters.

References

[1] I. T. U. (ITU), "Imt towards 2030 and beyond." [Online] https://www.itu.int/en/ITU-R/study-groups/rsg5/rwp5d/imt-2030/pages/default.aspx . Accessed: August 20, 2025.

[2] 3GPP, "The rel-20 work plan." [Online] https://www.3gpp.org/specifications-technologies/releases/release-20 . Accessed: August 20, 2025.

[3] Y. Liu, Y. He, Y. Lin, and L. Tang, "Toward native artificial intelligence in 6g," in *2022 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, pp. 1–6, IEEE, 2022.

[4] C. Sun, T. Cui, W. Zhang, Y. Bai, S. Wang, and H. Li, "On the combination of ai and wireless technologies: 3gpp standardization progress," in *2024 IEEE/CIC International Conference on Communications in China (ICCC Workshops)*, pp. 523–528, IEEE, 2024.

[5] S. Majumdar, Q. Wei, S. Schwarzmann, R. Trivisonno, and G. Carle, "Towards ai-native 6g systems: Standards enablers for 6g network automation," *TechRxiv Preprint*, 2024. Online https://doi.org/10.36227/techrxiv.24025986.v1.

[6] Y. Jeon and S. Pack, "Hierarchical network data analytics framework for 6g network automation: Design and implementation," *IEEE Internet Computing*, vol. 28, no. 2, pp. 38–46, 2024.

[7] L. F. R. Moreira, R. Moreira, F. d. O. Silva, and A. R. Backes, "Towards cognitive service delivery on b5g through aiaas architecture," in *Anais do Workshop de Redes 6G (W6G)*, SBC, 2024.

[8] Y. Lu, Y. Liu, K. Zhang, H. Ji, and V. C. M. Leung, "Federated learning-empowered mobile network management for 5g and beyond networks: From access to core," *IEEE Communications Surveys Tutorials*, vol. 26, no. 1, pp. 110–143, 2024.

[9] P. Li, Y. Xing, and W. Li, "Distributed ai-native architecture for 6g networks," in *2022 International Conference on Information Processing and Network Provisioning (ICIPNP)*, pp. 57–62, IEEE, 2022.

[10] A. Nadar and J. Härri, "Enhancing network data analytics functions: Integrating aiaas with ml model provisioning," in *2024 2nd Mediterranean Communication and Computer Networking Conference (MedComNet)*, pp. 1–6, IEEE, 2024.

[11] P. Li and Y. Xing, "Always-on personal ai for 6g networks," in *2024 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, pp. 1–6, IEEE, 2024.

[12] C. Yeh, Y.-S. Choi, Y.-J. Ko, and I.-G. Kim, "Standardization and technology trends of artificial intelligence for mobile systems," *Computer Communications*, vol. 213, pp. 169–178, 2024.

[13] X. Lin, M. Chen, H. Rydén, J. Jeong, H. Lee, M. Sundberg, R. Timo, H. S. Razaghi, and H. V. Poor, "Fueling the next quantum leap in cellular networks: Embracing ai in 5g evolution towards 6g," *IEEE Communications Magazine (submitted)*, 2021. arXiv:2111.10663.

[14] F. Matera, M. Settembre, A. Detti, A. Vizzarri, F. Mazzenga, D. Ronzani, J. Llorca, A. Tulino, and S. Barbarossa, "Opportunities and challenges for the integration of ai in network evolution toward 6g," *Proceedings of the AEIT International Annual Conference*, 2024.

[15] W. Yi, Y. Fu, J. Cao, L. Gan, L. Xiong, and H. Li, "Towards seamless 6g and ai/ml convergence: Architectural enhancements and security challenges," *IEEE Network*, 2024.

[16] S. Tarkoma, R. Morabito, and J. Sauvola, "Ai-native interconnect framework for integration of large language model technologies in 6g systems," *arXiv preprint arXiv:2311.05842*, 2023.

[17] "3GPP ts 29.510 v18.1.0: 5g system; application layer functions; service-based architecture (sba) web services," Tech. Rep. Release 18, 3rd Generation Partnership Project (3GPP), December 2024. URL: https://www.3gpp.org/DynaReport/29510.htm.

[18] L. Lamport, *Specifying systems*, vol. 388. Addison-Wesley Boston, 2002.

[19] B. Samanta, "Stability analysis of discrete-time systems," in *Introduction to Digital Control: An Integrated Approach*, pp. 159–194, Springer, 2024. Chapter 7: Stability Analysis of Discrete-Time Systems.

[20] J. M. de Oliveira, "Repository: A protocol-based framework for distributed ai lifecycle management in 6g via nwdaf." [Online] https://github.com/juniamaisa/AIasServ-LifeCycle/tree/main . Accessed: June 25, 2025.

[21] B. Alpern and F. B. Schneider, "Defining liveness," *Information processing letters*, vol. 21, no. 4, pp. 181–185, 1985.

[22] Istio Project, "Istio Microbenchmarks." https://github.com/istio/istio/wiki/Microbenchmarks, 2025. Accessed: June 29, 2025.

[23] "3GPP TR 23.288: Enhanced Support for Edge Applications," tech. rep., 3GPP, Mar. 2025. Version 18.0.0.

[24] MTLab, "onnx2caffe: Convert ONNX models to Caffe." https://github.com/MTLab/onnx2caffe/tree/master, 2025. Accessed: June 29, 2025.

[25] J. M. Oliveira, J. Almeida, E. De Britto e Silva, L. F. Rodrigues Moreira, R. Moreira, F. O. Silva, D. F. Macedo, and J. M. Nogueira, "Anomaly detection employing a 5g core data analytics framework," in *2024 IEEE 13th International Conference on Cloud Networking (CloudNet)*, pp. 1–9, 2024.