Forecasting Before Acting: Recent Advances in Learning-Based Traffic Prediction and Intelligent Control in Open RAN

Thanh Thien-An Dang[a], Dongwook Won[a], Juyoung Kim[a], Viet Anh-Huy Le[a], Sungrae Cho[a,*]

[a]*Department of Computer Science and Engineering, Chung-Ang University, Seoul 06974, South Korea*

Abstract

Open Radio Access Network (O-RAN) disaggregates the RAN into standardized units coordinated by Near-Real-Time (Near-RT) and Non-Real-Time (Non-RT) RAN Intelligent Controllers (RICs), enabling closed-loop optimization. Proactive control requires accurate prediction of traffic demand, user equipment (UE) mobility, and resource consumption before congestion or service level agreement violations occur. Yet machine learning (ML)-based proposals remain fragmented across granularities, interface placements, and evaluation environments, with no unified framework for comparing designs or assessing evidence quality.

This paper surveys recent works on ML-based traffic and UE behavior prediction that explicitly feed O-RAN control loops. We classify each work along four axes: (i) ML tier, separating architectures with structural inductive bias from standard deep learning and classical methods; (ii) traffic target at UE, slice, and cell granularities; (iii) evaluation environment across live deployments, emulation, and simulation; and (iv) O-RAN interface placement across A1, E2, O1, and O2.

Our analysis reveals four headline findings. First, a *forecast-then-act* pipeline, composing a predictor with a downstream deep reinforcement learning or optimization controller, dominates closed-loop design across granularities. Second, standard deep learning predominates, whereas architectures with structural inductive bias remain a minority. Third, evaluation is dominated by custom, unstandardized simulators, with no surveyed work performing live closed-loop actuation on an operational deployment. Fourth, open-source reproducibility is near-absent, with only one complete release of code, real data, and evaluation scripts. We then identify open challenges in per-UE forecasting, standardized data exposure, and open-source evaluation infrastructure, and outline a research roadmap toward 6G predictive intelligence in O-RAN.

Keywords: Open RAN, traffic prediction, machine learning, RAN Intelligent Controller, resource management, 6G

1. Introduction

The radio access network (RAN) is undergoing a fundamental architectural shift. Open RAN (O-RAN) disaggregates monolithic base stations into open, standardized components connected through well-defined interfaces, exposing Near-Real-Time (Near-RT) and Non-Real-Time (Non-RT) RAN Intelligent Controllers (RICs) that host data-driven control applications [1]. These controllers, operating across Near-RT (10 ms to 1 s) and Non-RT (more than 1 s) control loops, can in principle act on predictions of traffic demand, mobility events, and resource consumption before congestion or Service Level Agreement (SLA) violations occur. Proactive control of this kind requires accurate, timely forecasts: an xApp that predicts the throughput demand of each user equipment (UE) 1–5 s ahead can pre-allocate physical resource blocks (PRBs) before the surge arrives; an rApp that forecasts cell-aggregate load over the next 10–60 s can solve a constrained optimization and commit a resource policy over the A1 interface before the planning window expires. Without reliable forecasts, control decisions remain reactive and forfeit the proactive potential that open interfaces make possible [2].

Machine learning (ML) has become the dominant approach for RAN traffic prediction, and the O-RAN com-

*Corresponding author

Email addresses: attdang@uclab.re.kr (Thanh Thien-An Dang), dwwon@uclab.re.kr (Dongwook Won), jykim@uclab.re.kr (Juyoung Kim), lvahuy@uclab.re.kr (Viet Anh-Huy Le), srcho@cau.ac.kr (Sungrae Cho)

munity has produced a rapidly growing body of proposals. In addition to explicit traffic forecasting, we also consider adjacent learning-based works whose outputs support proactive O-RAN control, such as mobility prediction, anomaly detection, and slice-aware classification. These works, however, differ along dimensions that make direct comparison difficult: the granularity of the prediction target (per-UE, per-slice, or cell-aggregate), the O-RAN interface that carries prediction inputs and outputs (E2, A1, O1, or O2), the ML tier of the model (novel architecture, standard deep learning, or classical/shallow learning), and the realism of the evaluation environment (live testbed, emulation, or simulation). Among the 21 technical papers surveyed here, none validate through live closed-loop actuation on an operational O-RAN deployment, only five (about a quarter) evaluate on real-world operator traces, and around 62% rely solely on simulation or synthetic traffic traces. Only one paper releases a complete open-source implementation together with real-traffic data [3]. Roughly one-third of the surveyed works (seven of 21) place per-UE inference inside the Near-RT RIC, but all of them subscribe to per-UE counters via the E2SM-KPM Periodic Report style; none yet exploit the Measurement Event style introduced in E2SM-KPM v7.0 [4, 5], leaving sub-interval reactive per-UE control as the open frontier.

The practical stakes of getting this right are concrete and operator-facing. Accurate forecasts let the RIC preallocate slice resources before demand surges [6, 7], anticipate handovers before a UE crosses a cell boundary [8, 9], flag per-UE anomalies inside the Near-RT loop budget [10], and sleep lightly loaded cells to save energy without breaching SLAs [11]. Treating these use cases as a single prediction-for-control problem is therefore not only operationally valuable but scientifically clarifying, because it exposes the shared evidence gaps that an interface-aware survey can diagnose once and for all.

No prior survey structures these works jointly by O-RAN interface placement, ML paradigm, traffic target, and evaluation fidelity. Existing reviews either focus on a single granularity, treat O-RAN as a general AI/ML deployment target without distinguishing interface-level constraints, or cover only a narrow technology window. This gap makes it difficult for practitioners to identify which architectural choices are well-supported by evidence, which remain open research problems, and where testbed evaluation is still lacking.

This paper addresses that gap. We survey 25 papers (21 technical works and 4 related surveys) published between 2023 and 2026, classify each along a four-axis taxonomy, and synthesize cross-cutting findings. Our specific contributions are:

- A four-axis taxonomy classifying 21 technical papers by ML tier, traffic target (UE, slice, or cell), evaluation environment, and O-RAN interface placement, consolidated in table 2.
- Identification of a recurring *forecast-then-act* pipeline pattern across all three granularities, in which a prediction head supplies input state to a downstream Deep Reinforcement Learning (DRL) or optimization controller via E2 control messages or A1 policies.
- Cross-cutting analysis of evaluation realism, open-source posture, and statistical rigor across the surveyed corpus, revealing structural gaps between reported accuracy and real-world deployability.
- An eight-challenge research roadmap spanning near-term engineering bottlenecks (event-driven per-UE xApps, real-world evaluation scarcity, open-source xApp reference repository, statistical rigor) and longer-horizon frontiers (multi-step long-horizon prediction, foundation models for RAN, privacy, fairness, and governance, multi-objective conflict-aware control).

Figure 1 gives a visual roadmap of the survey, whose remaining sections proceed in three stages. The *foundations* stage establishes the background: section 2 positions this work against prior O-RAN surveys, and section 3 describes the O-RAN architecture and AI/ML framework. The *survey core* stage carries the review itself: section 4 presents the survey methodology and classification taxonomy, and section 5 reviews the literature across UE-level, slice-level, and cell-level granularities. The *synthesis and outlook* stage interprets the corpus and looks ahead: section 6 presents cross-cutting analysis, section 7 identifies open challenges and a research roadmap, and section 8 concludes the paper. Abbreviations used throughout the paper are collected in the appendix.

2. Related Surveys

Several O-RAN survey papers have appeared in recent years, each targeting a distinct facet of the architecture. Table 1 summarizes how these works relate to the present paper along four criteria: existence of a prediction taxonomy, treatment of cross-granularity analysis, coverage of control loop design, and evaluation fidelity,

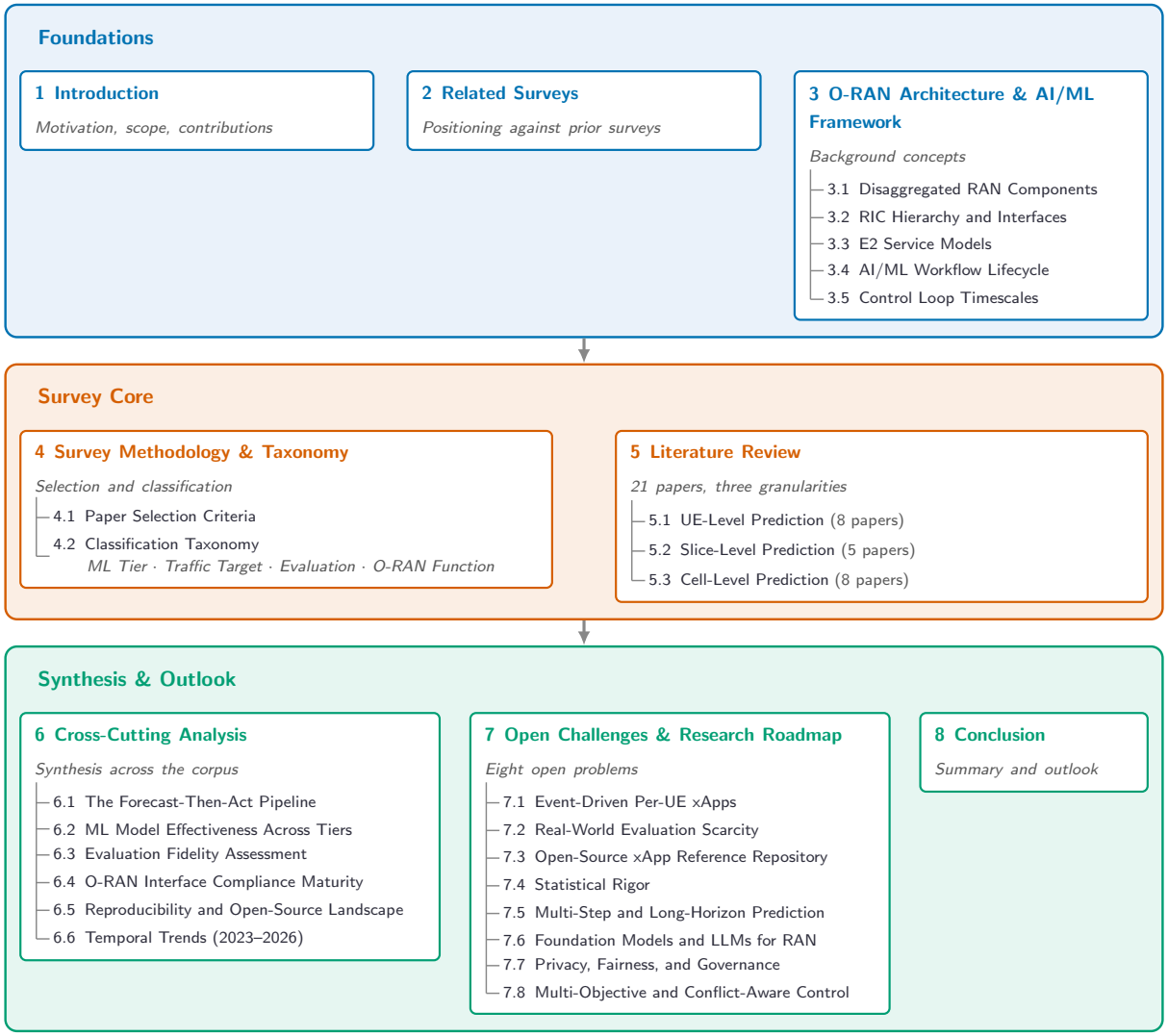

Figure 1: Navigation map of the survey. The eight sections are organized into three thematic stages. *Foundations* (Sections 1 to 3) covers motivation, related surveys, and O-RAN background. *Survey Core* (Sections 4 and 5) presents the methodology, the classification taxonomy, and the literature review. *Synthesis & Outlook* (Sections 6 to 8) provides cross-cutting analysis, identifies open challenges, and concludes. Each section lists its subsections as a directory tree, and the vertical arrows indicate the reading order across the stages.

alongside a descriptive scope column. None of the four prior surveys satisfies all four criteria simultaneously; this work is designed to fill that combined gap.

Polese et al. [1] provide the canonical O-RAN tutorial, covering the full specification stack, all interfaces (E2, A1, O1, O2, open fronthaul), the AI/ML deployment workflow, security threats, and experimental platforms. It is the primary architectural reference for the field. However, the paper does not offer a prediction taxonomy, does not classify ML models by technique tier, and does not analyze the fidelity of evaluations (simulation versus emulation versus real deployment). Every paper in this survey builds on the architecture described there; this work complements it by characterizing what prediction algorithms operate within that architecture and how rigorously they are evaluated.

Alam et al. [2] extend the architectural treatment to network slicing, covering deployment scenarios, service management and orchestration (SMO), transport-network slicing, and O-Cloud management up to O-RAN Release 004. Prediction appears as a motivating concept, namely the network slice subnet instance

(NSSI) resource-allocation optimization use case, but is not analyzed in detail: no taxonomy of ML models for traffic prediction is provided, and cross-granularity analysis (UE-level vs. slice-level vs. cell-level predictors) is absent. This survey fills that gap by classifying how prediction granularity determines both which O-RAN interface carries the predictor output and what control actions become feasible.

Elyasi et al. [12] survey xApp architectures, development techniques, AI/ML deployment scenarios, state-of-the-art use cases including traffic steering and resource allocation, and security challenges in the Near-RT RIC. The scope is xApp-centric and also covers quality of service (QoS) and anomaly detection use cases. Yet the paper does not provide a taxonomy of prediction models grouped by ML technique tier, does not address the fidelity of the evaluation evidence, and does not analyze cross-granularity prediction pipelines. The present work is complementary: where Elyasi et al. characterize the xApp ecosystem broadly, this paper focuses on the subset of works that implement explicit forecast-then-act pipelines and evaluates their evidence quality.

Mthethwa et al. [13] review ML techniques for O-RAN network slicing, proposing a functionality-driven taxonomy (forecasting, admission control, resource allocation, security) and highlighting hybrid Long Short-Term Memory (LSTM) plus DRL approaches. The scope is restricted to the slice granularity: UE-level and cell-level prediction are outside its remit, and all reviewed studies are simulation-based. The authors acknowledge that real-world testbed validation is the primary gap in the field. This paper addresses that gap directly by classifying evaluation fidelity across all granularities and identifying the minority of works that validate on real or emulated O-RAN deployments.

Taken together, the four prior surveys provide strong foundations in O-RAN architecture, slicing, xApp engineering, and ML technique selection, but leave four gaps unaddressed: (i) a structured prediction taxonomy classifying ML models by technique tier (novel architecture, standard deep learning, or classical/shallow learning); (ii) cross-granularity analysis spanning UE, slice, and cell levels; (iii) explicit analysis of how prediction granularity maps onto O-RAN interfaces and control loop timescales; and (iv) a systematic comparison of evaluation fidelity across the literature. As the final row of table 1 makes explicit, no prior O-RAN survey jointly classifies the field along interface placement, ML technique tier, traffic prediction target, and evaluation fidelity at once; each earlier work covers at most a subset of these axes. This survey is the first to address all four together.

3. O-RAN Architecture and AI/ML Framework

3.1. Disaggregated RAN Components

O-RAN disaggregates the traditional monolithic base station into four standardized functional units connected via open interfaces [1]. Closest to the antenna, the *O-Radio Unit* (O-RU) implements the low physical layer (low-PHY), handling frequency-time domain conversion operations: Inverse Fast Fourier Transform (IFFT) and cyclic prefix (CP) insertion in the downlink, and FFT and CP removal in the uplink, along with digital beamforming. It also hosts the radio frequency (RF) frontend, which drives over-the-air transmission and reception. The O-RU connects to the O-DU via the *Open Fronthaul* (Open FH) interface, which carries four functional planes: control, user, synchronization, and management (C/U/S/M). The C- and U-planes are encapsulated in enhanced Common Public Radio Interface (eCPRI) or IEEE 1914.3 payloads, the S-plane relies on Precision Time Protocol (PTP) or Physical Layer Frequency Signals (PLFS) for sub-microsecond synchronization, and the M-plane runs NETCONF over SSH/TLS for secure management [1].

The *O-Distributed Unit* (O-DU) implements the high physical layer (high-PHY), Medium Access Control (MAC), and Radio Link Control (RLC) sublayers [1, 14]. Within these layers, the O-DU handles time-sensitive functions such as PRB scheduling and Hybrid Automatic Repeat reQuest (HARQ) retransmission for attached UEs, among other procedures. The O-DU connects upward to the O-CU via the 3GPP F1 interface and is an *E2 node*: it exposes internal RAN state to the Near-RT RIC through the standardized E2 interface and accepts control commands in return.

The *O-Central Unit* (O-CU) handles the highest-layer RAN protocols and splits into two logical entities [2]. The *O-CU Control Plane* (O-CU-CP) implements Radio Resource Control (RRC) and the control-plane portion of the Packet Data Convergence Protocol (PDCP), managing UE attachment, mobility signaling, and bearer establishment. The *O-CU User Plane* (O-CU-UP) carries user-plane PDCP and the Service Data Adaptation Protocol (SDAP), transporting user traffic between the 5G core and the O-DU. Separating O-CU-CP and O-CU-UP allows independent scaling and enables per-slice dedication of user-plane resources: a single shared O-CU-CP may coexist with multiple dedicated O-CU-UP instances, one per network slice. Both O-CU-CP and O-CU-UP are E2 nodes and expose RAN telemetry and control hooks to the Near-RT RIC.

Table 1: Comparison of O-RAN surveys positioning this work. *Scope* describes the primary technical focus of each survey. *Prediction Taxonomy* indicates whether a structured classification of prediction models by ML technique tier (novel architecture, standard deep learning, or classical/shallow learning) is provided. *Cross-Granularity* indicates whether UE-level, slice-level, and cell-level prediction are analyzed together. *Control Loop* indicates whether the mapping of predictors to O-RAN interface and timescale is covered. *Eval Fidelity* indicates whether evaluation evidence is classified by deployment environment (simulation, emulation, real testbed).

Ref	Year	Scope	Prediction Taxonomy	Cross-Granularity	Control Loop	Eval Fidelity
[1]	2023	Full O-RAN spec stack, interfaces, security	No	No	Partial	No
[12]	2025	xApp architecture, AI/ML scenarios, security	No	No	Partial	No
[13]	2025	ML for slice management (slice-only)	Partial	No	No	No
[2]	2026	Slicing-aware O-RAN, deployment, SMO	No	No	Partial	No
This work	2026	Learning-based prediction across all granularities	Yes	Yes	Yes	Yes

The functional boundary between the O-DU and O-RU follows the *3GPP Option 7.2x* split, which places the partition between high-PHY at the O-DU and low-PHY at the O-RU. This partition places frequency-time domain conversion at the O-RU (IFFT/FFT, CP insertion/removal) and frequency-domain symbol processing at the O-DU (modulation, layer mapping, resource element mapping) [1]. The 7.2x split imposes strict fronthaul latency constraints, typically below 100 μs one-way, and demands multi-gigabit-per-second transport capacity, tightly coupling the O-RU to geographically proximate O-DU deployments.

3.2. RIC Hierarchy and Interfaces

O-RAN introduces two RAN Intelligent Controllers (RICs) that form a hierarchical, two-timescale closed-loop control architecture [1, 2]. Figure 2 illustrates the full component and interface topology.

The *Near-RT RIC* operates at control-loop periods of 10 ms to 1 s. It hosts *xApps*: containerized microservices that subscribe to E2 node telemetry, execute AI/ML inference, and issue control commands to the RAN. The O-RAN specifications define an xApp through a descriptor and a software image but do not mandate a specific orchestration substrate; in the O-RAN Software Community (OSC) reference implementation, xApps are packaged as Docker images and orchestrated as microservices on a Kubernetes cluster [1, 12]. Each xApp accesses live RAN state through the Shared Data Layer (SDL), subscribes to specific E2 nodes via the Subscription Manager, and translates model outputs into E2 control actions defined by E2 Service Models (see section 3.3). Because multiple xApps may simultaneously target the same E2 node, the Near-RT RIC includes a Conflict Mitigation module to arbitrate competing control requests [12].

The *Non-RT RIC* operates at periods greater than 1 s and resides within the *Service Management and Orchestration* (SMO) framework. It hosts *rApps*: applications responsible for long-horizon policy analysis, AI/ML model training, and enrichment information generation. rApps access SMO platform services through the *R1 interface*, which provides service registration, data exposure, and proxied access to the A1, O1, and O2 services. The Non-RT RIC owns the full AI/ML model lifecycle: collecting training data, training models offline, validating them, and deploying them to the Near-RT RIC [2].

The *A1 interface* connects the Non-RT RIC to the Near-RT RIC and carries three categories of messages, each delivered by a dedicated A1 service [1, 15]. First, the *A1 policy management* service, abbreviated *A1-P*, delivers *policy guidance* that instructs xApps on optimization objectives (e.g., prioritize URLLC throughput, apply energy-saving thresholds); a policy is expressed as a JSON object whose schema is defined per use case. Second, the *A1 enrichment information* service (*A1-EI*) provides external context not available inside the RAN, such as predicted traffic load trajectories or location-based demand estimates. Third, the *A1 ML model management* service transfers trained model parameters or

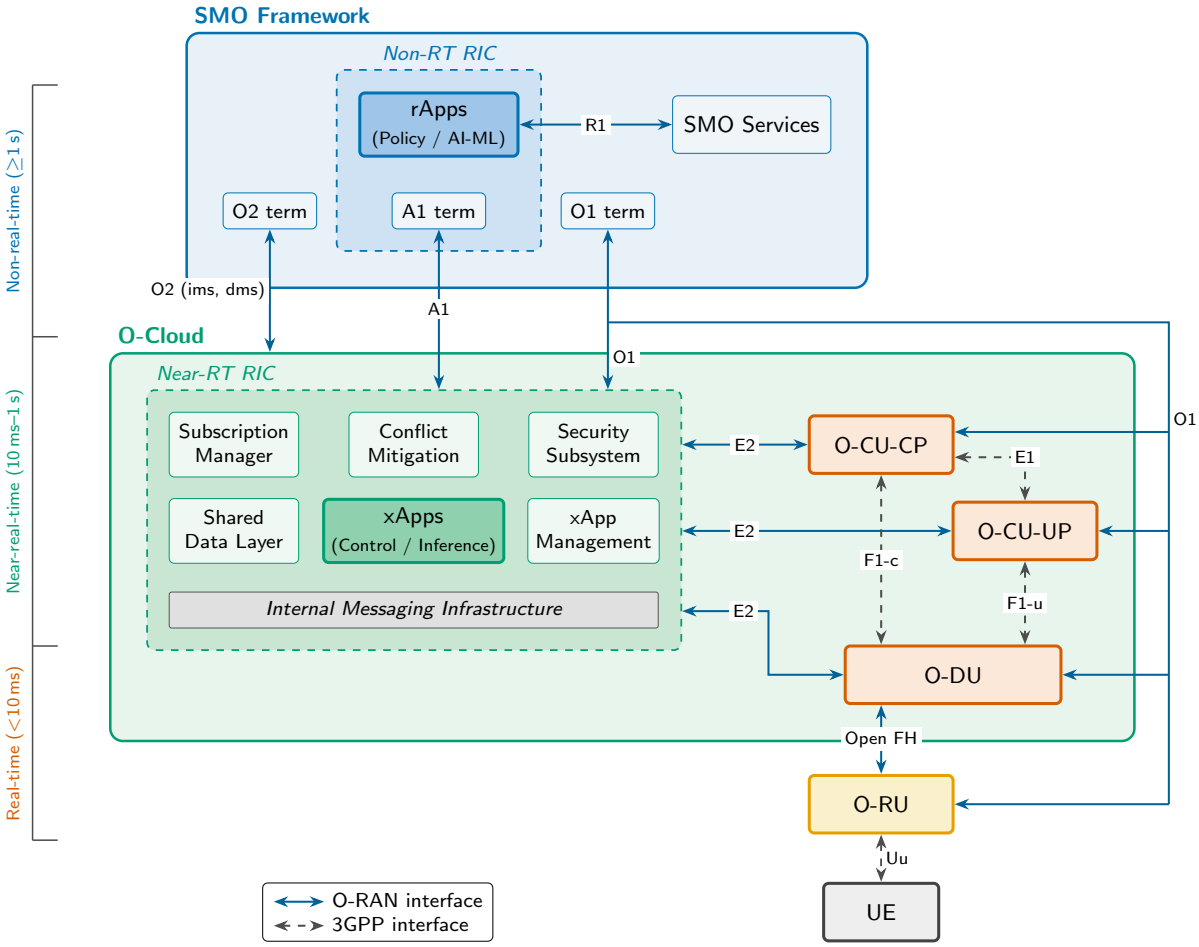

Figure 2: O-RAN reference architecture. Virtualized network functions (Near-RT RIC, O-CU-CP, O-CU-UP, O-DU) are deployed on the O-Cloud, while the O-RU remains a physical network function at the cell site. The Non-RT RIC hosts rApps and the A1 termination, residing inside the Service Management and Orchestration (SMO) framework alongside shared SMO services and the O1/O2 terminations. The Near-RT RIC hosts xApps together with platform services (Shared Data Layer, Subscription Manager, Conflict Mitigation, Security Subsystem, and xApp Management) on an internal messaging infrastructure. Solid arrows denote O-RAN-defined interfaces (R1, A1, O1, O2, E2, Open Fronthaul); dashed arrows denote 3GPP-defined interfaces (E1, F1-c, F1-u, Uu). E2 carries KPM telemetry upward and control actions downward between the Near-RT RIC and each E2 node (O-CU-CP, O-CU-UP, O-DU). A1 carries policy guidance (A1-P), enrichment information (A1-EI), and AI/ML model management from the Non-RT RIC to the Near-RT RIC. Based on Polese et al. [1].

deployment configurations. Through A1-EI, a prediction rApp can relay traffic forecasts directly to an xApp, implementing the forecast-then-act pipeline at the interface level [1].

The *O1 interface* connects the SMO to all managed O-RAN elements, including O-CU, O-DU, and the Near-RT RIC, and carries Fault, Configuration, Accounting, Performance, and Security (FCAPS) management functions over NETCONF/YANG and REST/HTTPS. For AI/ML, O1 is the primary channel through which long-period performance counters (hourly PRB utilization, UE count per cell, handover (HO) statistics) flow into the Non-RT RIC data lake for offline model training.

The *O2 interface* connects the SMO to the *O-Cloud*, the virtualization platform hosting O-RAN network functions as containerized workloads. O2 provides two sub-interfaces: O2ims for infrastructure resource management (compute, storage, and network capacity) and O2dms for network function lifecycle management (deployment, scaling, and termination of containerized functions). For AI/ML workflows, O2 enables the SMO to provision compute resources for model training and to deploy updated model containers to the Near-RT RIC

following a retraining cycle.

3.3. E2 Service Models

The *E2 Application Protocol* (E2AP) defines four abstract service types: REPORT (periodic or event-triggered telemetry streaming), INSERT (event-driven data injection into the xApp), CONTROL (direct parameter commands to the E2 node), and POLICY (autonomy rules enforced at the E2 node without per-event round trips). Concrete data formats and semantics for these services are specified by *E2 Service Models* (E2SMs), which define what an xApp can observe and what it can actuate at each E2 node [1].

E2SM-KPM (Key Performance Measurement) defines the reporting interface for RAN telemetry collection. KPM measurement containers carry per-UE and per-cell indicators including downlink and uplink throughput, PRB utilization, buffer occupancy, Reference Signal Received Power (RSRP), Modulation and Coding Scheme (MCS), Channel Quality Indicator (CQI), packet latency, and session counts. E2SM-KPM supports both periodic reporting (configurable period) and event-triggered reporting (the Measurement Event style, added in v7.0, which emits a report when a subscribed measurement crosses an operator-defined condition). This lets xApps configure the measurement granularity and reporting period to match their prediction model's input requirements. For traffic prediction, E2SM-KPM defines the observable feature space: only metrics exposed through this service model are available to an xApp at near-real-time timescales.

E2SM-RC (RAN Control) defines the actuation interface for xApp-initiated RAN parameter changes. It enables xApps to trigger UE-level and cell-level control actions such as HO initiation, PRB allocation adjustments, scheduling policy selection, and beam management parameter updates. In the forecast-then-act pipeline, E2SM-RC is the mechanism through which a prediction output becomes a concrete RAN control command: the xApp sends an E2SM-RC CONTROL message to an O-DU or O-CU-CP to enforce its scheduling or mobility decision.

E2SM-CCC (Cell Configuration and Control) addresses cell-level parameter configuration, enabling xApps to modify transmission power levels, antenna tilt and weight configurations, and cell activation or deactivation. E2SM-CCC complements E2SM-RC by covering configuration-scope actions that affect all UEs within a cell simultaneously, rather than individual UE-level decisions, and is particularly relevant for energy-saving and coverage optimization use cases.

E2SM compliance is critical for xApp portability across multi-vendor deployments. An xApp developed against a specific E2SM version can operate over any conformant E2 node regardless of vendor implementation, enabling the O-RAN open marketplace model in which third-party applications control multi-vendor RAN infrastructure [2, 12].

3.4. AI/ML Workflow Lifecycle

O-RAN Working Group 2 (WG2) specifies a standardized AI/ML workflow that governs how data-driven models are developed, validated, deployed, and maintained within the O-RAN architecture [1, 16]. The workflow comprises six sequential steps spanning both the Non-RT and Near-RT RICs.

Step 1: Data collection and preparation. Data are gathered over the O1, A1, and E2 interfaces and stored in a centralized data lake repository within the SMO [1]. A preparation phase shapes and formats samples for the target model, applying normalization, scaling, reshaping, and optional dimensionality reduction (e.g., autoencoders).

Step 2: Training. Training is performed offline at the Non-RT RIC or an external ML platform. O-RAN specifications prohibit the deployment of any untrained model; online training is permitted only to fine-tune a model previously trained offline [1, 16].

Step 3: Validation and publishing. Trained models are validated against unseen datasets to verify reliability and robustness under diverse traffic and deployment conditions. Models that pass validation are published to an AI/ML catalog on the SMO/Non-RT RIC; failed models return to redesign and retraining.

Step 4: Deployment. Validated models in the catalog are delivered to the inference host over the O1 interface, following either image-based deployment (containerized xApp/rApp) or file-based deployment (standalone file executed in an inference environment). WG2 defines five deployment scenarios that place training, inference, and continuous-operation blocks at different points between the SMO/Non-RT RIC and the Near-RT RIC [16].

Step 5: AI/ML execution and inference. Deployed models consume live data to produce classifications, predictions, or control decisions. At the Near-RT RIC, an xApp ingests E2SM-KPM reports and issues E2SM-RC control commands within the 10 ms–1 s loop; at the Non-RT RIC, an rApp generates A1 policies or enrichment information at periods exceeding 1 s.

Step 6: Continuous operations. The Non-RT RIC continuously verifies, monitors, and analyzes deployed

models, and recommends refinement or retraining when accuracy, SLA compliance, or input distributions degrade [16]. Updated models re-enter the workflow at Step 2 without service interruption. Figure 3 illustrates this complete lifecycle.

The forecast-then-act pipeline maps directly onto Steps 1–5 of this workflow: a model trained on historical KPMs (Steps 1–2) is validated and published (Step 3), deployed to a Near-RT xApp (Step 4), and consumes live E2SM-KPM inputs to issue E2SM-RC commands within the same inference cycle (Step 5). Step 6 closes the outer loop through continuous operations that monitor forecast accuracy and trigger retraining when degradation is detected, ensuring the prediction model remains calibrated to evolving traffic patterns.

3.5. Control Loop Timescales

O-RAN defines three control loop timescales that correspond to distinct optimization scopes, RAN components, and AI/ML model constraints [1, 2]. The appropriate timescale for a prediction-driven application determines where the prediction model runs, what features it can consume, and what control actions are available to it.

The *non-real-time* loop (period ≥1 s) is executed by rApps in the Non-RT RIC. At this timescale, rApps perform long-horizon traffic trend analysis, AI/ML model training, network-wide energy management, and slice SLA assurance. Works in this survey that operate primarily at this timescale include rApp-based probabilistic load forecasters [17], cell-level traffic predictors fed via O1 KPI streams [18–21], and a federated learning framework for distributed CPU demand provisioning [22].

The *near-real-time* loop (period 10 ms to 1 s) is executed by xApps in the Near-RT RIC, consuming E2SM-KPM reports and issuing E2SM-RC or E2SM-CCC commands. Works targeting this timescale include DRL-based slicing xApps [7, 23, 24], LSTM-based traffic steering and load balancing [6, 25], HO prediction xApps [8, 9, 26], anomaly detection for UE throughput [10], and traffic classification xApps [3, 27, 28]. Several works combine both timescales by deploying a prediction rApp at the Non-RT RIC that relays A1 enrichment information to a control xApp at the Near-RT RIC, exemplifying the forecast-then-act pattern [6, 11].

The *real-time* loop (period <10 ms) is handled internally by layer-1 and layer-2 functions within the O-DU and O-RU. Typical decisions at this timescale include scheduling, beam management, and feedback-less detection of physical-layer parameters such as MCS and interference [1]. ML-based traffic prediction cannot operate at this timescale: the latency budget precludes E2 round trips and RIC involvement. The O-RAN Alliance has proposed *dApps* (distributed applications) co-located within the O-DU or O-CU as a future extension for sub-10 ms AI-driven control, though this concept remains in early standardization and is outside the scope of the works surveyed here [1].

Table 2 catalogues all 21 technical works surveyed in this paper. Although the table does not include an explicit timescale column, the *App* column encodes control-loop placement: entries marked `xA` (xApp) target the near-real-time loop (10 ms–1 s) in the Near-RT RIC, entries marked `rA` (rApp) target the non-real-time loop (≥1 s) in the Non-RT RIC, and entries marked `xA+rA` span both timescales, typically via an A1-coupled forecast-then-act pipeline. With the O-RAN architecture and AI/ML framework established, section 4 formalizes how we selected these works and the four-axis taxonomy used to classify them.

4. Survey Methodology and Taxonomy

4.1. Paper Selection Criteria

Papers were retrieved from four databases: IEEE Xplore, ACM Digital Library, arXiv, and Google Scholar. The search combined the mandatory term *O-RAN* with at least one of *traffic prediction*, *forecasting*, *machine learning*, *xApp*, or *rApp*, covering publications from January 2023 through March 2026. This window aligns with the period during which O-RAN Alliance specifications reached sufficient maturity (Release 003 onward) to support meaningful ML integration in the Near-RT and Non-RT RICs.

Inclusion required three conditions. First, the paper must explicitly target the O-RAN architecture: papers proposing ML for generic 5G RAN without reference to O-RAN functional units or interfaces were excluded. **Second, the work must either (i) explicitly predict or forecast a traffic, mobility, or resource metric, or (ii) provide a learning-based classification, detection, or control component that directly supports proactive or closed-loop O-RAN control. Accordingly, the corpus includes both explicit forecast-then-act studies and adjacent learning-based works that inform or contextualize predictive O-RAN control.** Third, the paper must propose or empirically evaluate at least one ML model; purely analytical or protocol-design works were not included.

Exclusion criteria eliminated three categories of work. Architecture-only papers that describe O-RAN

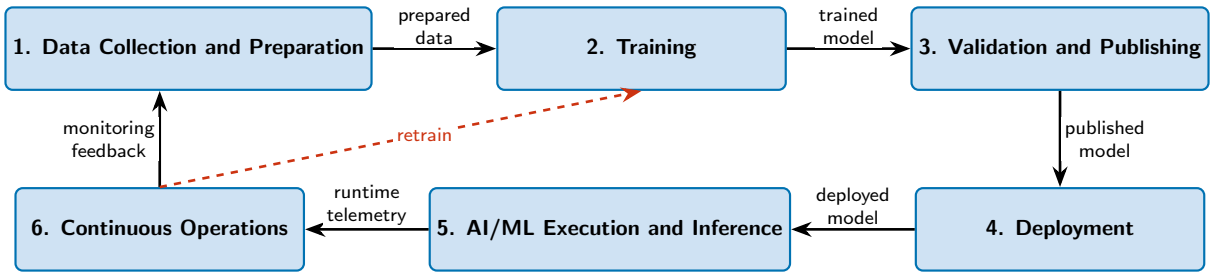

Figure 3: AI/ML workflow lifecycle in O-RAN. The six-step workflow comprises (1) data collection and preparation, (2) offline training, (3) validation and publishing to the AI/ML catalog, (4) deployment to xApps/rApps, (5) online execution and inference, and (6) continuous operations that monitor, verify, and trigger retraining of deployed models. Based on Polese et al. [1] and O-RAN Working Group 2 [16].

design without any ML prediction component are covered in section 2 as related surveys and are not counted in the technical corpus. Papers predating the O-RAN Alliance specification era (pre-2019) were excluded because they cannot address the standardized interface constraints (E2, A1, O1, O2) that define the present survey's scope. Non-peer-reviewed works without an arXiv preprint were excluded; conference papers with venue verification issues were retained only when a preprint was publicly available.

The screening followed a four-step funnel. The keyword query was first run across the four databases to retrieve an initial candidate pool. Titles and abstracts were then screened against the three inclusion conditions, records duplicated across databases were merged, and the surviving candidates underwent full-text eligibility assessment to confirm both the O-RAN-control focus and the presence of an evaluated ML model. We note that the intermediate tallies at each step (raw retrieval count, post-deduplication count, and post-screening count) were not logged during the search, so only the final corpus size is reported.

The process yielded 25 papers in total. The technical corpus is deliberately kept to 21 works because the strict O-RAN-control inclusion criterion, requiring both an explicit O-RAN functional-unit or interface target and a learning-based predictive or control component, excludes the far larger body of generic 5G-RAN ML literature, and because O-RAN-specific predictive control is still an emerging field whose feasible publication window opened only when the specifications matured (the 2023 to 2026 span noted above). Studies near this inclusion boundary were not excluded automatically. Such a study was included when it informed proactive or closed-loop O-RAN control, and excluded only when it had no O-RAN coupling. Four of the 25 are survey papers [1, 2, 12, 13] that provide architectural context and are analyzed separately in section 2 and table 1. The remaining 21 papers constitute the technical corpus analyzed in sections 5 and 6 and catalogued in table 2.

4.2. Classification Taxonomy

Works in the corpus are classified along four independent axes: *ML Tier*, *Traffic Target*, *Evaluation Paradigm*, and *O-RAN Function and Interface*. The axes are designed to be orthogonal: assignment along one axis does not constrain or imply assignment along another. Figure 4 visualizes the four-axis structure and the paper counts per category.

Axis 1: ML Tier

The ML tier axis groups models by architectural sophistication, following a three-level hierarchy.

Tier 1 (Advanced Models, 5 papers) encompasses architectures that employ full self-attention forecasting models or end-to-end graph representation learning on explicit relational structure. Transformer-based predictors include the Autoformer [11], PatchTST [18], and a Transformer used within a hierarchical DRL scheduler [29]. Graph neural models appear as a Variational Graph Autoencoder (VGAE) and SEAL [30] and a Hypergraph Graph Neural Network (GNN) [27]. The boundary rule for Tier 1 is strict: a Graph Attention Network (GAT)-Gated Recurrent Unit (GRU) hybrid such as T-GAT [19] is classified as Tier 2 because the dominant learning mechanism is a recurrent network with attention as an auxiliary aggregator, rather than a full self-attention architecture or a message-passing GNN trained end-to-end on graph structure.

Tier 2 (Deep Learning, 11 papers) covers standard recurrent and convolutional architectures, probabilistic forecasters built on recurrent backbones, plus DRL-based control policies. The LSTM family includes

Table 2: Detailed comparison of 21 technical works on learning-based traffic prediction in O-RAN. Organized by ML Tier (Tier 1=advanced, Tier 2=deep learning, Tier 3=classical). *Target*: UE=user-equipment-level, Sl=slice-level, Ce=cell-level. *Eval*: RW-T=real-world trace, RW-L=real-world live (none observed), Em=emulation, Sim=simulation. *App*: xA=xApp, rA=rApp. *OS*: Y=open-source code released.

Ref	Year	Target	Eval	Key Model	Interface(s)	App	OS
Tier 1: Advanced (Transformers, GNNs)							
[29]	2026	UE	Sim	Transformer + PPO (Hier-DRL)	Conceptual	xA+rA	N
[27]	2026	UE	RW-T	Tri-HyperGNN-Sketch	E2	xA+rA	Y
[30]	2025	UE	RW-T	VGAE + SEAL (GNN link pred.)	O1, A1, E2	rA	N
[11]	2024	Ce	Sim	Autoformer (TTI-level pred.)	E2, A1, O1	xA+rA	N
[18]	2024	Ce	RW-T	PatchTST / DLinear + XAI	O1, A1	rA	N
Tier 2: Deep Learning (LSTM/GRU, DRL, Federated NN, CNN)							
[8]	2026	UE	Sim	S-LSTM (transfer learning, HO)	E2 (KPM+RC)	xA	N
[17]	2025	Ce	Sim	DeepAR (GluonTS)	O1	rA	N
[7]	2025	Sl	Em	LSTM + DDPG (dual-timescale)	A1, E2	xA+rA	N
[9]	2025	UE	RW-T	3-layer LSTM (HO classifier)	E2 (KPM)	xA	Y
[3]	2024	Sl	Em	CNN (traffic slice classifier)	E2	xA	Y
[23]	2024	Sl	Em	PPO (latency-aware slicing)	E2, A1	xA	N
[22]	2024	Ce	Sim	FL + LTN (neuro-symbolic)	E2, A1, O1	xA	N
[6]	2023	Sl	Sim	LSTM + SAC (dist. DRL)	A1*, E2*	xA+rA	N
[24]	2023	Sl	Sim	PPO + Forecast-aided DRL	A1	xA	N
[25]	2023	Ce	Sim	LSTM + K-means (cell throughput)	E2	xA	N
[19]	2023	Ce	RW-T	T-GAT (GRU + GAT hybrid)	Conceptual	xA	N
Tier 3: Classical ML (Tree-based, KNN, GNB, Tabular RL)							
[10]	2025	UE	Sim	Iso. Forest + Random Forest	E2, A1	xA	N
[21]	2025	Ce	Sim	XGBoost + grid search (tuned)	Conceptual	xA	N
[20]	2024	Ce	Sim	LSTM + XGBoost (benchmark)	Conceptual	xA	N
[28]	2024	UE	Sim	Tabular RL (multi-layer TS)	O1, A1, E2	xA	N
[26]	2023	UE	Sim	GNB + KNN (UE compatibility)	Conceptual	xA	N

* architectural role inferred; interface not named in paper

works targeting UE HO [8, 9], cell throughput [20, 25], slice demand [6, 7], and cell-level traffic [19]. Kasuluru et al. [17] propose a probabilistic forecasting rApp based on DeepAR and an adaptive percentile selection mechanism (DYNp). Although the study reports a Transformer estimator as a comparison baseline, its primary contribution is the DeepAR-based probabilistic pipeline, so it is classified as Tier 2. DRL-based xApps using Proximal Policy Optimization (PPO) [23, 24] and Soft Actor-Critic (SAC) [6] also fall in Tier 2, as do federated neuro-symbolic learning [22] and Convolutional Neural Network (CNN)-based classification [3].

Tier 3 (Classical ML, 5 papers) groups non-neural or shallow models. XGBoost is used for cell throughput regression [20, 21]; ensemble methods (Isolation Forest, Random Forest) serve anomaly detection [10]; Gaussian Naive Bayes (GNB) and k-Nearest Neighbors (KNN) are applied to UE compatibility classification [26]; and tabular multi-step RL [28] drives traffic steering without neural function approximation. Several papers span two tiers (e.g., LSTM+XGBoost in Nassar et al.); in such cases, the paper is assigned to the tier of its primary proposed model.

Axis 2: Traffic Target

Traffic target captures the *granularity* at which the ML model produces predictions, which in turn determines the O-RAN interface over which prediction outputs travel and the control actions they enable.

UE-level (8 papers) targets per-user metrics: HO events, link quality, compatibility time, or per-UE demand. UE-level outputs are most actionable for Near-RT RIC xApps because the E2 interface exposes per-UE identifiers (UEID) via E2SM-KPM. Works in this category span mobility prediction [8, 9, 30], link prediction for proactive HO and V2X compatibility-time [26, 27], anomaly detection for per-UE events [10], anti-jamming task scheduling [29], and intelligent traffic steering [28].

Slice-level (5 papers) forecasts aggregate resource demand per tenant or service class. Slice-level predictors naturally fit the Non-RT RIC rApp role because inter-slice allocation operates at timescales above one second; prediction outputs are carried as A1 policies to Near-RT RIC xApps for fine-grained enforcement [6, 7, 23, 24].

Cell-level (8 papers) models sector-wide aggregate load, throughput, or CPU demand. This category has

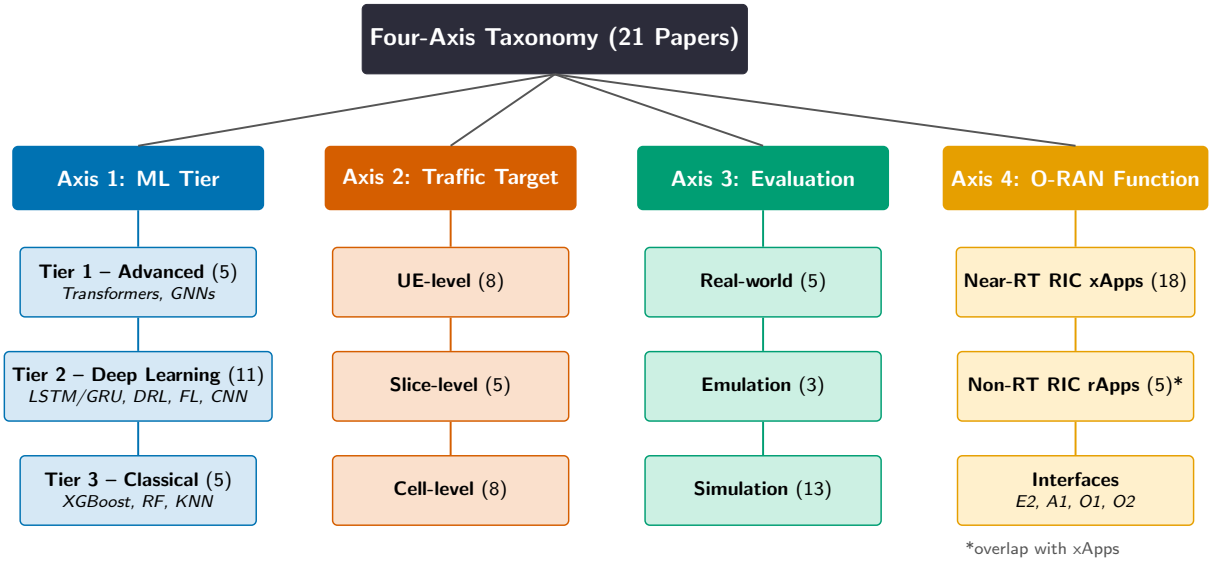

Figure 4: Four-axis taxonomy for classifying learning-based traffic prediction works in O-RAN. Axis 1 (ML Tier) spans Tier 1 (advanced: Transformers, GNNs) through Tier 3 (classical: tree-based, tabular RL). Axis 2 (Traffic Target) covers UE, slice, and cell granularities. Axis 3 (Evaluation) ranges from simulation to emulation to real-world deployments. Axis 4 (O-RAN Function) maps works to Near-RT/Non-RT RIC and standardized interfaces (E2, A1, O1, O2). Annotated with paper counts per category.

the longest history and the largest share of real-world evaluations [18, 19, 22, 25].

Axis 3: Evaluation Paradigm

The evaluation paradigm records the highest-fidelity environment in which the trained model was executed, forming a credibility gradient from simulation to real deployment. We assign each paper to one of three categories using the following priority rule, applied top-down. First, if the model executes inside a hardware-in-the-loop testbed (Colosseum, POWDER), the paper is *Emulation*, regardless of data origin. If instead the trained model is deployed in a live production network and issues closed-loop control commands (E2SM-RC scheduling, A1 policy push, or equivalent), the paper is *Real-world live* (RW-live). Conversely, if the paper's primary evaluation source (that is the dataset producing the headline metric) is real operator or network data evaluated offline, the paper is *Real-world trace* (RW-trace). In all other cases the paper is *Simulation*. Real data used only for training, with synthetic data carrying the headline evaluation, does not move a paper into Real-world.

Real-world (5 papers) draws on real operator or network data; all five fall in the RW-trace sub-category, and no paper in the corpus reaches RW-live, which we therefore retain in the taxonomy as the target state framing the open challenge in section 7.2. Within RW-trace we observe a provenance gradient. Dzaferagic et al. [9] and Henna and Rathnayake [27] perform *first-party live data collection* from O-RAN deployments: Dzaferagic et al. [9] subscribe a custom xApp on the Accelleran DRAX Near-RT RIC at the OpenIreland testbed to E2SM-KPM v2.0 with a one-second reporting period and accumulate an 11-hour, 40,000-sample handover dataset, while Henna and Rathnayake [27] use the Eurecom Elastic-Mon5G2019 dataset collected through the Elastic-Mon v0.1 framework on an OpenAirInterface RAN and core deployment. In both cases the trained model is then evaluated offline on a held-out split of the collected trace, with no closed-loop actuation back to the running network. The remaining three RW-trace papers rely on third-party historical operator datasets: Xiong et al. [19] train on 28 days of KPM data from 48 production base stations near Beijing, Pérez et al. [18] evaluate forecasting models on a three-month 4G production trace from a large European metropolitan operator, and Bermudez et al. [30] validate GNN link prediction on the anonymized Huawei open telecom graph at million-UE scale.

Emulation (3 papers) uses hardware-in-the-loop testbeds. Colosseum [3, 23] and POWDER with OpenAirInterface (OAI) [7] exercise live radio stacks and O-RAN software, providing higher fidelity than simula-

tion without requiring production deployments. Groen et al. [3] additionally capture real 5G traces on commercial smartphones with PCAPdroid and replay them in Colosseum, combining real-data origin with emulator execution.

Simulation (13 papers) encompasses custom Python or MATLAB environments and standard simulators (ns-3, EXata). Simulation enables rapid algorithmic iteration but cannot validate interface compliance, timing behavior under real channel conditions, or scalability to full gNB deployments. The dominance of simulation (62% of the corpus) is the most significant structural gap identified in this survey. Crucially, these simulators are overwhelmingly custom and unstandardized (bespoke Python or MATLAB scripts with no shared scenarios, topologies, or traffic models), so each paper evaluates on a private environment. This absence of a common simulation baseline directly undermines cross-paper comparability, because no two studies can be placed on the same footing.

Axis 4: O-RAN Function and Interface

The fourth axis records which RIC layer hosts the application (Near-RT, Non-RT, or both) and which standardized interfaces the work references or implements.

Near-RT RIC xApps dominate, appearing in 18 of 21 papers, of which 5 also involve a Non-RT RIC rApp through a rApp-to-xApp prediction pipeline. Interface compliance is assessed at three levels: *conceptual*, where the paper invokes interface names without implementing them; *partial*, where data collection via E2SM-KPM or policy passing via A1 is implemented but not fully conformant; and *full E2SM*, where the xApp subscribes to E2SM-KPM for data and issues E2SM-RC control messages. Only Panitsas et al. [8] explicitly targets both E2SM-KPM and E2SM-RC in a single predictive HO xApp, and only Dzaferagic et al. [9] validates full E2SM-KPM data collection from a commercial node. Details per paper are listed in table 2. Applying this taxonomy, section 5 now reviews the corpus organized by traffic-target granularity, proceeding from UE-level to slice-level to cell-level prediction.

5. Literature Review

Prior work on learning-based prediction in O-RAN spans three traffic granularities. *UE-level* works target per-user mobility, demand, and anomaly patterns; *slice-level* works forecast aggregate resource demand per tenant or service class; and *cell-level* works model sector-wide load and throughput. Each granularity motivates a distinct placement within the O-RAN control hierarchy: UE-level outputs are most actionable for Near-RT RIC xApps, while slice- and cell-level outputs naturally feed Non-RT RIC rApps or serve as auxiliary state for DRL controllers.

5.1. UE-Level Prediction

UE-level prediction covers three sub-tasks: HO and mobility prediction, link quality prediction for traffic steering, and anomaly or security-event detection. Several of these per-UE Near-RT RIC works are examined from an event-driven xApp perspective in our prior conference study [5]. Across the eight works surveyed here, LSTM-based sequence models (Tier 2) dominate the HO and anomaly sub-tasks, while three Tier 1 graph, hypergraph, and DRL designs (Henna and Rathnayake [27], Bermudez et al. [30], and Asemian et al. [29]) handle the cases requiring multi-node radio relationships. Real-world or emulation-based evaluation remains rare: only Dzaferagic et al. [9] validates on a commercial O-RAN testbed, Bermudez et al. [30] uses an industry-scale Huawei dataset, and Henna and Rathnayake [27] trains on real 4G/5G RAN traces. The remaining five works rely on simulation or synthetic data, leaving real-world validation the primary open gap. Table 3 summarizes the key dimensions of all eight works.

Handover and Mobility Prediction

Panitsas et al. [8] propose a scalable Encoder–S-LSTM–Decoder architecture for predictive HO in O-RAN. A three-layer multi-layer perceptron encoder (128 neurons) compresses a $K \times N$ RSRP measurement matrix into a 128-dimensional embedding, two stacked LSTM layers extract temporal dependencies, and a softmax decoder outputs per-BS association probabilities over W future steps. An xApp subscribes to per-UE RSRP via E2SM-KPM and dispatches HO commands via E2SM-RC, completing a closed control loop. Under 10-fold cross-validation on a 10 GB simulation dataset spanning $N \in \{21, 36, 51\}$ base stations, the model achieves >95% classification accuracy and reduces HO frequency by up to 46% relative to the 3GPP Event A3 baseline. A transfer learning module re-initialises only the Encoder input and Decoder output layers on topology change, cutting retraining time by at least 30% (up to 52%).

Dzaferagic et al. [9] provide one of the few HO prediction studies grounded in real O-RAN hardware. A three-layer LSTM (32/64/32 hidden units) is trained on 40,000 samples collected at 1 s granularity from the OpenIreland testbed (Accelleran DRAX Near-RT

Table 3: Detailed comparison of UE-level prediction works.

Ref	Sub-task	Model	Input KPMs	Horizon	Key Metric	Value	Dataset
[8]	HO prediction	Encoder–S-LSTM–Decoder	RSRP ($K\times N$ matrix, 50 ms)	Up to 3 s	Accuracy; HO reduction	>95%; 46%	Simulation (custom, 10 GB)
[9]	HO event prediction	3-layer LSTM	RSRP, RSRQ, SINR (serving + 3 nbrs)	9 s	Recall; cost reduction	88%; >80%	Real-world (OpenIreland, 40k)
[27]	Link quality / traffic steering	Tri-HyperGNN-Sketch	CQI per UE–RU link (>100 metrics)	Near-RT (10 ms)	Link pred. accuracy	99.99%	Real-world (ElasticMon5G2019)
[30]	Next-cell / HO target prediction	VGAE+GAT, SEAL	RSRP, RSRQ, MCS, RBs, signal power	<1 s (static)	F1 score	0.84	Real-world (Huawei + EXata synth)
[28]	Traffic steering (flow-split)	Tabular RL + convex opt.	Queue lengths, arrival rates, CSI	10 ms (1 frame)	Delay constraint	<10 ms	Simulation (Rician, 12 UEs)
[10]	Anomaly detection (UE throughput)	RF, IF, AE, AE-1SVM	PRB DL, RSRP, RSSINR, RSRQ	Per-report ($\ll$10 ms)	F1; nbr cell reduction	0.90; 41.27%	Simulation (OSC ric-app-ad, 10k)
[29]	Anti-jamming scheduling	ADPHRP-JE (PPO) + Transformer	Jamming prob. history (10 frames)	10 ms (1 frame)	Mis-predicted slots; drop ratio	13; 0.917	Simulation (MATLAB, synthetic)
[26]	V2X compatibility time	GNB, KNN, shallow NN	Distance, velocity, direction labels	Up to >15 s (binned)	Test accuracy	99.53%	Simulation (synthetic, 20k)

RIC, Benetel RU650 radio units, Cumucore 5G core, licensed 3.85–3.95 GHz spectrum). The input sequence spans a 10 s history window with 12 features per step: RSRP, Reference Signal Received Quality (RSRQ), and Signal-to-Interference-plus-Noise Ratio (SINR) for the serving cell plus three neighbours, collected via E2SM-KPM v2.0. At a 9 s horizon, the model achieves 88% recall and approximately 80% precision, and reduces dynamic spectrum-leasing cost by more than 80% relative to static long-term acquisition. Its cost function maps recall (missed HOs) and precision (false alarms) to operational expenditure, bridging ML classification metrics and business-level SLAs, a methodological novelty absent from the other works. A public dataset (Mendeley Data) supports reproducibility.

Link Quality Prediction and Traffic Steering

Henna and Rathnayake [27] address traffic steering in multi-connectivity O-RAN through a Tier 1 hypergraph neural network. Their Tri-HyperGNN-Sketch model constructs a UE–RU interference hypergraph from CQI measurements, applies probabilistic Tri-subgraph sampling to extract $O(|E|)$-complexity triangle-centric enclosing subgraphs, then performs L-layer hyperedge convolution with attention to predict binary link-steering decisions. Trained on the real-world Eurecom ElasticMon5G2019 dataset and validated over five splits with three random seeds ($p < 0.001$ paired t-test), the model achieves 99.99% link-prediction accuracy versus 95.1% for Tri-GNN-Sketch and 94.67% for SEAL, with an ablation confirming the hypergraph component contributes the largest gain (+4.89 percentage points). An SMO-hosted retraining loop and Near-RT RIC xApp complete the forecast-then-act O-RAN pipeline; code is publicly available.

Bermudez et al. [30] apply GNN-based link prediction to proactive HO target identification, deploying the model as an rApp in the Non-RT RIC. Two architectures are compared: VGAE with a GAT encoder (probabilistic, global-graph inference) and SEAL (subgraph extraction around candidate UE–cell pairs), evaluated on a synthetic 5G dataset from the Keysight EXata emulator (70 UEs, 31 cells) and a large-scale anonymised Huawei telecom graph (up to 1 M UEs and 20 k cells). Using only graph structure, the model achieves F1 = 0.84 on the real-world Huawei partition, surpassing a prior GNN+Transformer radio-link-failure baseline (F1 = 0.79). VGAE scales to 10^6 UEs with inference below 1 s, compliant with Non-RT RIC latency, though it degrades at very high node densities. Predicted target cells flow via the A1 interface to a Connection Management xApp that prepares the HO before UE metrics degrade.

Nguyen et al. [28] tackle traffic steering through a multi-layer optimisation framework that jointly optimises flow-split distribution, congestion control, and beamforming across the Non-RT and Near-RT RIC. A three-step tabular RL procedure (Tier 3) with exponentiated mirror descent learns per-UE flow-split fractions from one-frame-delayed queue-length and arrival-rate observations delivered via the O1 interface, while inner-

approximation and bisection algorithms solve the short-term congestion and beamforming sub-problems as Near-RT RIC xApps. Formal proofs bound the utility-optimality gap as $O(1/\varphi)$ and queue length as $O(\sqrt{\varphi})$, guarantees absent from learning-only approaches. The 10 ms delay bound holds for $\varphi \leq 25$ in simulation, and the framework outperforms single-layer baselines across antenna counts $M \in \{16, 128\}$.

Anomaly Detection and Security

Azkaei et al. [10] decompose UE-level anomaly detection into two algorithms: Algorithm 1 classifies serving-cell KPI reports (PRB utilization, RSRP, RSSINR, RSRQ), where RSSINR is the Received Signal Strength Indicator-to-Interference-plus-Noise Ratio, as degraded or normal to trigger HO recovery; Algorithm 2 filters neighbour cells (15 features: RSRP, RSSINR, and RSRQ per five candidate cells) to remove poor HO targets, reducing the candidate set by 41.27% on average. Across four models on the OSC `ric-app-ad` sample dataset (10,000 KPI reports, 20 UEs), Random Forest achieves F1 = 0.90 and 93% accuracy at 0.22 ms inference per 20 UEs; all four satisfy Near-RT RIC time budgets (maximum 2.49 ms). SHAP and permutation importance identify PRB utilization and RSSINR as the primary anomaly drivers, providing operator-interpretable root-cause analysis.

Asemian et al. [29] address a security-driven UE scheduling problem under ON-OFF jamming in an O-RAN deployment that integrates multi-access edge computing. A hierarchical DRL framework decomposes the partially observable Markov decision process into two agents: ADPHRP-JE, a PPO-based estimator that predicts per-slot jamming probability vectors one frame (10 ms) ahead from 10-frame histories of observed jamming states; and WMAC-TS, a Transformer-based actor-critic (128-unit embedding, three attention heads) that schedules tasks only to predicted safe slots. ADPHRP-JE accumulates only 13 mis-predicted slots over 100 steps, a 48% reduction over DDQN with historical data and 73% over standard DDQN. The Transformer scheduler keeps linear complexity $O(N_a)$, cutting execution time substantially over a genetic-algorithm baseline. The framework is scoped as an xApp/rApp in an AI-native RAN but validated in MATLAB simulation only.

V2X and Classical ML

Rastogi et al. [26] propose mapping compatibility-time prediction for vehicle pairs onto the two-level O-RAN RIC architecture (Non-RT and Near-RT). Three Tier 3 classifiers (Gaussian Naive Bayes, KNN, and a shallow NN) predict five discrete connectivity-duration classes from five engineered kinematic features (distance, relative velocity, direction labels, approach tendency) on a synthetic 20,000-sample vehicular dataset, with KNN best at 99.53% test accuracy. The O-RAN xApp integration is conceptual and unimplemented, with no HO KPIs reported. Despite its limited scope, the work illustrates the Non-RT RIC (offline training) to Near-RT RIC (online inference) deployment pattern, and motivates trajectory-aware sequence models as future extensions.

Several patterns emerge from this group. First, LSTM remains the workhorse for HO and anomaly sub-tasks, while Tier 1 architectures appear in the two traffic-steering works that model spatial UE–RU relationships (Henna and Rathnayake [27] and Bermudez et al. [30]) and in the hierarchical DRL security work (Asemian et al. [29]). Second, real-world validation is the exception: only Dzaferagic et al. [9] collects data from a licensed-spectrum commercial testbed with multi-vendor hardware, while Henna and Rathnayake [27] and Bermudez et al. [30] use real RAN traces but do not close the O-RAN control loop on live infrastructure, leaving the remaining five simulation-only works of uncertain transferability. Third, E2SM compliance is incomplete: only Panitsas et al. [8] (E2SM-KPM, E2SM-RC) and Dzaferagic et al. [9] (E2SM-KPM v2.0) exercise standardised service models, while most others rely on conceptual or non-standard surrogates. Finally, prediction horizons span four orders of magnitude, from per-report anomaly detection (sub-millisecond) through 9 s HO forecasting to vehicle compatibility durations exceeding 15 s, yet no work systematically studies how horizon length interacts with O-RAN control loop timescale and SLA tightness, an open question discussed in section 7.

5.2. Slice-Level Prediction

Slice-level works are grouped by the output granularity of the ML module, not by their input granularity. The five papers reviewed here produce slice-oriented outputs in three forms: forecast-aided slicing pipelines [6, 7, 24], traffic slice classification [3], and latency-aware reactive slicing [23]. Table 4 summarizes the five slice-level works.

Lotfi and Afghah [6] establish the canonical rApp–xApp template for the forecast-then-act pipeline at the slice granularity. A two-layer LSTM rApp in the Non-RT RIC is pre-trained on an 18-step look-back window of per-slice UE density and traffic load; its one-step-ahead predictions are appended to the Soft Actor-Critic (SAC) xApp state at every Near-RT RIC decision step.

In a simulated three-slice, seven-DU OFDMA network, the prediction-augmented agent achieves up to 7.7 % higher cumulative discounted reward and lower QoS-violation variance than DRL without forecasting, and represents, per the authors' claim, the first combination of a recurrent neural network (RNN) predictor with distributed DRL for O-RAN network slicing.

Alchaab et al. [7] extend this template to a two-timescale formulation validated on the NSF POWDER testbed with the OpenAirInterface 5G stack. The Long-Term Resource Allocation (LTRA) component, an LSTM rApp in the Non-RT RIC, forecasts average resource-block (RB) demand per slice over a long time interval (LTI); the Short-Term Resource Scheduling (STRS) component, a Deep Deterministic Policy Gradient (DDPG) xApp in the Near-RT RIC, refines allocations each short interval using LTRA predictions in its state vector. Trained on the Telecom Italia Milan cellular traffic dataset [31], the integrated variant (ILS-RLSlice) achieves a mean absolute error (MAE) of 0.0603 with 6.5 ms inference on real hardware, satisfying the sub-10 ms Near-RT RIC constraint, and attains a resource utilization ratio (RUT) of 0.9995 against a near-optimal target, substantially improving over the standalone actor-critic baseline.

Nagib et al. [24] move beyond state augmentation to analyze the mechanism by which forecasting improves DRL convergence. A Non-RT RIC forecasting module predicts per-slice traffic contribution $\hat{\kappa}$ over a horizon of up to ten slicing windows ($\leq$1 s) and translates it into a recommended PRB allocation. Via the A1 interface, the module applies *action distillation*: at each step it computes the Euclidean distance between the PPO xApp's intended action and the forecast-based allocation, and when the distance exceeds a 7 % threshold the agent takes a midpoint distilled action instead. Consulted on only 8.9 % of steps on average, the module delivers up to 86.3 % faster convergence and a 300 % increase in scenarios reaching near-optimal allocation versus standard DRL, with convergence maintained for Gaussian forecast error $\sigma \leq 0.25$. The authors reference a public code repository [24], though the link resolves to an empty project at the time of writing.

Groen et al. [3] address the prerequisite problem of identifying which slice a UE's traffic stream belongs to in real time. TRACTOR deploys a CNN xApp in the Near-RT RIC that polls 17 O-RAN-compliant KPIs via the E2 interface every 250 ms, stacks them into a sliding window matrix, and outputs a slice-type label (eMBB, URLLC, mMTC, or idle) per UE. Evaluated on Colosseum using real 5G traces replayed from 447 minutes of commercial smartphone captures, the system achieves over 95 % offline accuracy (window $T = 64$, 16 s context) and over 92 % in live xApp deployment, using no user-identifying fields. Because it labels current rather than future slice types, this is a classify-then-act contribution; the authors identify proactive slice-type prediction as future work. The complete toolchain, dataset, and xApp are open-source [3].

Raftopoulos et al. [23] address SLA-constrained PRB slicing without explicit traffic prediction. A PPO xApp trained offline on Colosseum collects RAN KPMs from the E2 interface and end-to-end latency enrichment information from the Non-RT RIC via A1, then outputs a per-slice PRB mask every 250 ms. Embedding the current SLA tolerance and latency threshold in the observation vector enables zero-shot adaptation to time-varying SLA requirements without retraining. In a dynamic SLA scenario, PPO achieves 8.3× fewer SLA violations and 60 % fewer PRB allocations than a Deep Q-Network (DQN) baseline (fixed 30 PRBs), showing that SLA-parameterised observation design is a strong reactive baseline. The model maps per-UE latency statistics to slice-level PRB allocations without generating a demand forecast.

Several cross-cutting observations emerge from this sub-literature. The three forecast+DRL works [6, 7, 24] share the rApp–xApp architectural split: the predictor resides in the Non-RT RIC at coarser timescales while the controller executes in the Near-RT RIC at sub-second granularity, a pattern formalised in section 6. Evaluation fidelity spans custom simulation [6, 24], emulation on Colosseum [3, 23], and testbed deployment with real traffic data [7], though the Telecom Italia dataset used by Alchaab et al. [7] dates from 2013–2014 Milan traffic [31], raising representativeness concerns for 5G profiles. The two non-forecasting works [3, 23] achieve strong near-term control but capture no demand trends over lookahead horizons, the gap the forecast-then-act designs address. Open-source posture is uneven: only Groen et al. [3] provides a complete public release (toolchain, author-collected dataset, and xApp), while Nagib et al. [24] references a repository that is empty on inspection.

5.3. Cell-Level Prediction

Among the three prediction granularities surveyed, cell-level forecasting is the most mature sub-field. It encompasses the largest share of real-world and emulation-based evaluations, benefits from decades of cellular traffic measurement infrastructure, and maps most directly to the Non-RT RIC rApp deployment pattern: models ingest performance-measurement and resource-utilization reports through O1 Performance

Table 4: Detailed comparison of slice-level works.

Ref	Approach	Model	Testbed	Pred. Horizon	Key Metric	Value
[6]	Forecast+SAC	LSTM rApp + SAC xApp	Python sim.	1 step	Reward gain	+7.7 %
[7]	Forecast+DDPG (dual-TS)	LSTM rApp + DDPG xApp	POWDER (OAI)	LTI (unspec.)	MAE / RUT	0.0603 / 0.9995
[24]	Action-distilled PPO	PPO xApp + forecast	Custom sim.	1–10 windows (≤1 s)	Convergence speed	+86.3 %
[3]	Slice classification	CNN xApp	Colosseum	N/A (classify)	Offline accuracy	>95 %
[23]	SLA-adaptive DRL	PPO xApp	Colosseum	N/A (reactive)	SLA violations	8.3× fewer

Assurance Management Services [32], then issue policies at Non-RT timescale [15, 17, 18]. The reviewed works span spatial-temporal graph learning, Transformer-based forecasting, probabilistic prediction, classical regression, and federated neuro-symbolic learning, establishing a clear progression from simple statistical baselines toward architectures that translate prediction accuracy into measurable network KPI gains such as energy efficiency and load balance. Table 5 compares the eight papers discussed below.

Spatial-Temporal Models

Xiong et al. [19] propose T-GAT, a GRU cell whose standard linear projections are replaced by Graph Attention Network (GAT) layers, enabling simultaneous aggregation of spatial and feature-domain dependencies at each recurrent step. It is trained and evaluated on a real operator dataset of 48 base stations in a Beijing district over 28 consecutive days (1-hour granularity, 8 KPMs including PRB utilization, RRC connections, and downlink traffic), with multi-feature spatial graphs built offline via Pearson correlation and dynamic time warping to give separate adjacency matrices per KPM. T-GAT achieves root mean squared error (RMSE) 0.1005 on normalized data, a 10% reduction over a GRU baseline and 8% over LSTM. This 28-day trace is one of the largest single-operator evaluation datasets in the corpus and anchors the cell-level sub-field's claim to high evaluation fidelity. Deployment is framed conceptually as a Non-RT RIC rApp, with no O-RAN interface integration implemented.

Transformer-Based Forecasting

Habib et al. [11] present the first reported application of a Transformer family model (Autoformer) to wireless traffic prediction at TTI (1 ms) granularity inside an O-RAN Near-RT RIC. The Autoformer xApp predicts aggregate cell traffic one step ahead and compares it to two thresholds: exceeding a high threshold activates a DQN-based traffic-steering xApp, while falling below a low threshold activates a DQN-based cell-sleeping rApp on the Non-RT RIC. This prediction-led on-demand orchestration avoids the conflicts of running both applications simultaneously. Over a 48-hour MATLAB 5G Toolbox simulation with 5 cells and 60 UEs, the scheme achieves 39.7% higher average energy efficiency than always-on traffic steering and 10.1% higher throughput than always-on cell sleeping. This energy-efficiency gain is the strongest single-metric improvement reported at cell level in the survey and shows that prediction accuracy translates directly into network KPI gains.

Pérez et al. [18] systematically benchmark DLinear, PatchTST, and LSTM on two real-world production 4G network traces: a 3-month European metropolitan dataset of aggregate cell traffic and a passive-monitoring dataset of RRC-connected user counts. At 10-minute granularity with multi-step horizons up to 60 steps (10 hours), DLinear dominates single-step prediction while PatchTST surpasses both alternatives for multi-step traffic volume forecasting (MAE 0.179 vs. LSTM 0.182 vs. DLinear 0.217). An XAI pipeline based on SHAP relevance scores and Gramian Angular Summation Fields explains the gap: DLinear relies on a sparse set of high-relevance input samples and struggles with infrequent weekend patterns, whereas PatchTST distributes attention across the full history window. Both models are cast as rApps in the Non-RT RIC, collecting KPMs via O1 and issuing guidance via A1. The study identifies horizon length as a primary axis of model selection, an insight missing from most comparative surveys.

Probabilistic Forecasting

Kasuluru et al. [17] present the first rApp deployment of probabilistic forecasting in a multi-tenant O-RAN scenario. Rather than a single point estimate, the system trains three probabilistic models (DeepAR, Transformer, and SFF) to output a full distribution of 99 percentiles per time step for cumulative PRB demand across all tenants. A novel adaptive algorithm, DYNp, selects the appropriate percentile each step based on whether the prior allocation caused an SLA violation,

Table 5: Detailed comparison of cell-level prediction works.

Ref	Approach	Model	Dataset / Testbed	KPM Predicted	Key Result
[19]	Spatial-temporal GNN	T-GAT (GAT-GRU hybrid)	Real operator traces, 48 BSs, Beijing, 28 days	8 KPMs (PRB util., traffic, users)	RMSE 0.1005; −10% vs. GRU
[11]	Transformer + on-demand orchestration	Autoformer xApp + DQN rApp	MATLAB 5G simulation, 5 cells, 60 UEs	Aggregate cell traffic (Mbps)	+39.7% energy efficiency vs. always-on steering
[18]	Transformer / linear benchmark	PatchTST, DLinear, LSTM	Real 4G prod. traces, European metro area	Traffic volume; RRC user count	PatchTST MAE 0.179; DLinear best single-step
[17]	Probabilistic forecasting + adaptive allocation	DeepAR + DYNp (rApp, Non-RT RIC)	Simulation, 50 BSs, Milan, 10 weeks hourly	Cumulative multi-tenant PRBs	DeepAR+DYNp: SLA violation 8%, over-prov. 0.509
[25]	K-means + LSTM xApp	LSTM (2-layer, 57-cell output)	Custom 3GPP UMa sim., 570 UEs / 57 PCIs	Per-cell DL throughput	Cells at target load: 14.04% → 35.09%
[20]	Classical benchmark	LSTM vs. XGBoost	Synthetic 5G NWDAF dataset, 5 cells	Cell throughput (Gbps)	LSTM MSE 0.2075 vs. XGBoost 0.4322
[21]	Hyperparameter-tuned XGBoost	XGBoost (grid search)	Synthetic 3GPP-inspired dataset, 5 cells, 6 months	Cell throughput (Gbps)	$R^2 = 99.5\%$; MSE 0.2895 (−33% vs. default)
[22]	Federated neuro-symbolic AI	FLMR (FL + Logic Tensor Network)	Real O-RAN testbed traces (IEEE DataPort)	vBS CPU utilization	6× lower provisioning error vs. DeepCog FL

forwarding the selected PRB count to the O-DU via O1. On a simulated Milan urban network (50 base stations, 15,000 users, 10 weeks of hourly data), DeepAR achieves the best point accuracy at the 50th percentile (mean squared error (MSE) 0.006 vs. LSTM 53.055, MAE 0.297 vs. LSTM 5.770), and combined with DYNp reduces the SLA violation rate to 8% while keeping over-provisioning volume at 0.509, the best joint result tested. Static percentile selection cannot minimize both metrics at once: the 99th percentile eliminates SLA violations but incurs over-provisioning volume of 1.05 (DeepAR) to 9.19 (SFF). This is the sole probabilistic forecaster in the corpus to provide distributional outputs for risk-aware PRB provisioning within the Non-RT RIC.

Load Balancing

Ntassah et al. [25] combine K-means UE clustering with a two-layer LSTM (200/100 neurons, 57-cell output) to predict per-cell DL throughput over a custom 3GPP TR 38.901 Urban Macro simulator with 570 UEs across 57 PCIs. Gating HO decisions on predicted throughput differentials rather than RSRP alone raises the percentage of cells at the target load of 10 UEs from 14.04% to 35.09% (a 149% increase), showing that a throughput-forecast xApp can improve load balance without explicit DRL.

Classical Benchmarks

Nassar et al. [20] benchmark LSTM against XGBoost for one-week-ahead cell throughput prediction on a synthetic 5G dataset (5 cells, 15-minute granularity). LSTM achieves MSE 0.2075 versus XGBoost 0.4322, a 52% reduction, at roughly three orders of magnitude longer training time (minutes vs. milliseconds). The paper frames prediction as a precursor to proactive HO management in O-RAN but implements no xApp. Nassar et al. [21] extend this work by applying grid-search hyperparameter tuning to XGBoost on the same dataset, testing squared-error and gamma objective functions. The tuned squared-error model achieves $R^2 = 99.5\%$ and MSE 0.2895, a 33% reduction over default parameters, and is containerized with FastAPI and Docker as an early xApp-integration prototype. Together, these two works show that systematically tuned gradient-boosted trees can rival an untuned LSTM on synthetic cell data, though real-world validation on operator traces is absent from both.

Federated Learning

Roy et al. [22] address vBS CPU utilization prediction in a shared O-Cloud environment hosting 50 virtual base station instances. Their FLMR (Federated Machine Reasoning) framework fuses Federated Learning (FL) with Logic Tensor Networks (LTN), replacing conventional regression loss with a logical-constraint satisfaction level. Trained on real O-RAN testbed trace data (Salvat et al., IEEE DataPort) in simulation, FLMR reduces combined over- and under-provisioning error by a factor of 6 relative to the DeepCog FL baseline while achieving approximately 98% logical constraint satisfaction. The paper introduces the dApp concept, co-located at vBS instances for tight-timescale inference, extending the O-RAN control hierarchy beyond standard Near-RT RIC timescales. This is the only neuro-symbolic approach reviewed at cell level and the only

work to prioritize interpretability as a primary design objective alongside accuracy.

Cell-level prediction stands out from its UE-level and slice-level counterparts in the breadth and fidelity of its evaluation settings: the Beijing operator traces in Xiong et al. [19] and the 4G production traces in Pérez et al. [18] provide real-world grounding absent from the simulation-heavy UE-level literature. The clearest evidence that prediction accuracy translates into operational gains comes from Habib et al. [11], whose 39.7% energy-efficiency improvement comes not from tuning a single application but from using a Transformer forecast to decide which application runs, the most direct mapping of cell-level forecasting to rApp deployment at Non-RT timescales. Kasuluru et al. [17] extend this paradigm: rather than a binary control action, DYNp selects a percentile from a probabilistic forecast distribution, providing risk-aware PRB provisioning beyond static thresholds or point-estimate models. The FLMR framework of Roy et al. [22] adds privacy and interpretability, increasingly relevant as operators deploy shared O-Cloud infrastructure across tenants. Remaining gaps at cell level include multi-site generalization beyond single-operator or single-city datasets, closed-loop validation on standards-compliant testbeds, and longitudinal stability of models trained on weeks of data. Having reviewed each work in isolation, section 6 now reads across the corpus to surface the pipeline patterns, evaluation-fidelity gaps, and reproducibility shortfalls that recur regardless of granularity.

6. Cross-Cutting Analysis

The preceding three subsections examined the 21 surveyed papers tier by tier. This section synthesises observations that cut across all granularity levels, organising them around six dimensions: the forecast-then-act pipeline, ML model effectiveness, evaluation fidelity, O-RAN interface compliance, reproducibility, and temporal trends. Table 6 provides a quantitative reference for papers where directly comparable metrics are available.

6.1. The Forecast-Then-Act Pipeline

Every paper in this survey either predicts a future network state before issuing a control action, reacts purely to the current state, or classifies the current state without forecasting. To formalise this, let $f\colon \mathcal{X} \to \hat{\mathcal{Y}}$ denote the forecasting model mapping input features X to a predicted output $\hat{Y}$, and let $\pi\colon \hat{\mathcal{Y}} \to \mathcal{A}$ denote the control policy mapping that prediction to an action A (e.g., PRB allocation, HO command, slot assignment); the forecast-then-act pipeline composes the two as $a_t = \pi(f(x_t))$.

Three structural variants appear across the 21 papers. In **V1 (state augmentation)**, the forecast $\hat{Y}$ is concatenated with the current environment state S and fed into a DRL agent that still selects the action. Lotfi and Afghah [6] and Alchaab et al. [7] both couple an LSTM traffic predictor with a DRL slice manager, reporting a 7.7% reward gain over prediction-free DRL and a 97.9% MAE reduction over an actor-critic baseline, respectively. Nagib et al. [24] use a more refined mechanism: the forecast generates a suggested PRB allocation compared with the DRL agent's proposed action via Euclidean distance, and if the gap exceeds a threshold a distilled midpoint action replaces the agent's choice (consulted in only 8.9% of steps on average), yielding up to 86.3% faster DRL convergence. Asemian et al. [29] feed per-slot jamming probability forecasts from their ADPHRP-JE estimator directly into the state space of a Transformer-based task scheduler, restricting its action to predicted safe slots.

In **V2 (threshold gating)**, the forecast $\hat{Y}$ is compared against a fixed threshold whose outcome activates or deactivates a downstream application rather than shaping its action space. Habib et al. [11], the single V2 instance, deploy an Autoformer xApp that predicts aggregate cell traffic and applies two thresholds (T_{hp}, T_{ht}) to gate a traffic-steering xApp or a cell-sleeping rApp, achieving 39.7% higher energy efficiency than always-on traffic steering.

In **V3 (direct action)**, the forecast $\hat{Y}$ is the action, or is mapped to one through a lightweight rule without a DRL stage. Nine papers instantiate V3. Panitsas et al. [8] apply argmax over predicted per-BS probability distributions to select the HO target directly. Dzaferagic et al. [9] map a binary HO-event prediction to a spectrum-leasing decision. Henna and Rathnayake [27] and Bermudez et al. [30] translate link-quality predictions into traffic-steering and HO-preparation commands. Rastogi et al. [26] map a predicted compatibility-time class to a proactive HO trigger. Ntassah et al. [25] use predicted average cell throughput to select HO target cells. Kasuluru et al. [17] implement DYNp, selecting a percentile from a probabilistic forecast distribution as the PRB allocation, with the percentile adjusted on SLA violation feedback. Xiong et al. [19] produce one-step-ahead KPM forecasts for direct input to planning algorithms. Roy et al. [22] provision CPU resources directly from predicted vBS utilization, reducing combined over- and under-

Table 6: Cross-paper quantitative comparison for papers with directly comparable metrics. Eval: RW = real-world, Em = emulation, Sim = simulation. All metric values are as reported in the respective papers.

Ref	Year	Target	Metric	Value	Eval
UE-level classification					
[27]	2026	Link quality (UE–RU)	Accuracy	99.99%	RW
[26]	2023	V2X compat. time	Accuracy	99.53%	Sim
[8]	2026	HO target	Accuracy	>95%	Sim
[9]	2025	HO event	Recall	88%	RW
[30]	2025	Next serving cell	F1	0.84	RW
[10]	2025	Throughput anomaly	F1	0.90	Sim
Cell-level prediction error					
[17]	2025	PRB demand (DeepAR)	MSE	0.006	Sim
[17]	2025	PRB demand (LSTM)	MSE	53.055	Sim
[18]	2024	Traffic vol. (PatchTST)	MAE	0.179	RW
[18]	2024	Traffic vol. (LSTM)	MAE	0.182	RW
[19]	2023	8 KPMs (T-GAT)	RMSE	0.1005	RW
[19]	2023	8 KPMs (LSTM)	RMSE	0.1093	RW
[20]	2024	Cell throughput (LSTM)	MSE	0.2075	Sim
[21]	2025	Cell throughput (XGBoost)	MSE	0.2895	Sim
System-level KPI					
[8]	2026	HO frequency reduction	vs. 3GPP A3	46%	Sim
[9]	2025	Spectrum cost reduction	vs. long-term	>80%	RW
[11]	2024	Energy efficiency gain	vs. always-on	+39.7%	Sim
[24]	2023	DRL convergence rate	vs. no forecast	+86.3%	Sim
[6]	2023	Cumulative slice reward	vs. DRL w/o pred.	+7.7%	Sim
[23]	2024	SLA violations	vs. DQN	8.3× lower	Em
[17]	2025	SLA violation rate	DeepAR+DYNp	8%	Sim
[25]	2023	Cells at load target	before/after HO	14%→35%	Sim
[29]	2026	Task drop ratio	vs. GA	0.917 vs. 0.942	Sim
[22]	2024	Provisioning error	vs. DeepCog FL	6× lower	Sim

provisioning by 6×.

The remaining seven papers fall outside the forecast-then-act taxonomy: Nguyen et al. [28] execute reactive tabular RL on delayed queue-state observations; Azkaei et al. [10] perform detect-then-alert anomaly classification without multi-step-ahead prediction; Groen et al. [3] classify current traffic slice type (classify-then-act); Raftopoulos et al. [23] apply reactive DRL on current observations; and Pérez et al. [18], Nassar et al. [20], and Nassar et al. [21] provide forecasting modules without closing the control loop in their evaluations. In summary, V1 accounts for 4 papers, V2 for 1, V3 for 9, and 7 are reactive or incomplete pipelines.

6.2. ML Model Effectiveness Across Tiers

Following the tier taxonomy defined in section 4, the 21 papers distribute as: Tier 1 (novel architectures with structural inductive bias) = 5, Tier 2 (standard deep learning applied to O-RAN contexts) = 11, Tier 3 (classical ML or tabular RL in constrained settings) = 5.

Tier 1 papers [11, 18, 27, 29, 30] dominate problems with rich spatial or temporal structure: hypergraph topology over multi-RU fronthaul, graph-structured UE–cell association, long-horizon periodic traffic decomposition, and adversarial non-stationary jamming. Henna and Rathnayake [27] achieve 99.99% link-prediction accuracy, a 4.99 percentage-point gain over the best graph baseline (S-VGAE at 95%) and 8.17 percentage points over standard GCN, while Habib et al. [11] report 23.35% higher throughput and 22.54% higher energy efficiency than LSTM-based prediction.

Tier 2 papers retain the LSTM or DRL backbone in O-RAN contexts where temporal sequential modelling suffices. The sharpest intra-Tier 2 com-

parison comes from Kasuluru et al. [17]: on one dataset, DeepAR achieves MSE = 0.006 whereas LSTM achieves MSE = 53.055 (over 8800-fold), showing that within Tier 2 the model family matters substantially. Pérez et al. [18] add a complementary insight on horizon length: at a single step DLinear (Tier 3) has the lowest error, but on multi-step traffic volume prediction over the same dataset PatchTST (Tier 1) leads with MAE = 0.179 and MSE = 0.062, ahead of LSTM (MAE = 0.182, MSE = 0.076) and DLinear (MAE = 0.217, MSE = 0.106).

Tier 3 papers [10, 21, 22, 26, 28] remain competitive in short-horizon, latency-critical scenarios. Azkaei et al. [10] measure Random Forest inference at 0.22 ms for 20 UEs, within the 10 ms Near-RT RIC budget, and Rastogi et al. [26] achieve 99.53% KNN classification accuracy for V2V compatibility-time prediction, showing classical ML suffices when the feature space is compact. The overall pattern: Tier 1 is most advantageous when graph or long-range temporal structure is present, while Tier 3 remains viable for short-horizon, feature-lean, latency-critical tasks. One caveat governs every comparison above. The headline figures (accuracy, reward, F1, MSE, throughput gain, and similar) are not directly comparable across studies, since each work uses a different dataset, traffic scenario, train-test split, and metric definition. They characterize each method within its own evaluation setup rather than on a common yardstick, so within-paper comparisons (such as Kasuluru et al. [17] or Pérez et al. [18] contrasting models on one shared dataset) carry the strongest evidential weight.

6.3. Evaluation Fidelity Assessment

We assess evaluation environments along four fidelity tiers, assigned by a strict priority rule based on where the trained model is executed. A model run inside a hardware-in-the-loop testbed (Colosseum or POWDER) is Em, regardless of data origin; a model issuing closed-loop control commands in a live production network is RW-live; a model evaluated offline on real operator or network traces is RW-trace; all remaining cases are Sim (software simulators, synthetic or 3GPP datasets, MATLAB, ns-3, EXata). Auxiliary real data used only for training does not move a paper into a real-world tier.

Of the 21 papers, 13 rely on simulation, 5 on real-world data, and 3 on emulation. This simulation-heavy majority reflects the difficulty of obtaining O-RAN testbed time or live operator data. All five RW papers sit in the RW-trace sub-category; none reach RW-live, since no surveyed paper closes the loop with E2SM-RC or A1 policy actuation against a running production deployment during evaluation. Within RW-trace, two papers perform first-party live data collection from O-RAN deployments and three rely on third-party historical operator datasets. Dzaferagic et al. [9] collect 40,000 samples at one-second granularity from a commercial O-RAN deployment on the OpenIreland testbed (Accelleran DRAX Near-RT RIC, Benetel RU650 radio units, licensed 3.85–3.95 GHz spectrum) and demonstrate a greater-than-80% operational cost reduction. Henna and Rathnayake [27] use the Eurecom Elastic-Mon5G2019 dataset, real 4G/5G RAN monitoring data collected through the Elastic-Mon v0.1 framework on an OpenAirInterface RAN and core deployment, and report 99.99% link-prediction accuracy on an 80/20 split of the trace. Bermudez et al. [30] validate GNN link prediction on the Huawei open telecom graph with up to one million UEs and 20,000 cells. Xiong et al. [19] train on 28 days of KPM data from 48 base stations near Beijing. Pérez et al. [18] use a three-month 4G production trace from a large European metropolitan area. A further limitation of the RW-trace tier is data recency: several traces predate widespread 5G O-RAN deployment, so their representativeness for current 5G traffic profiles is uncertain. The Pérez et al. [18] traces come from a production 4G network and an LTE passive-monitoring tool, and the Xiong et al. [19] dataset was collected near Beijing in 2018, both predating 5G New Radio operation. The three emulation papers (Groen et al. 3, Raftopoulos et al. 23, Alchaab et al. 7) use Colosseum and POWDER for hardware-in-the-loop RF realism; Groen et al. [3] additionally captures real 5G traces on commercial smartphones and replays them in Colosseum, combining real-data origin with emulator execution.

By target granularity, UE-level has 3 RW papers, cell-level has 2, and slice-level has none. This slice-level absence suggests slice prediction relies more heavily on simulation, possibly because multi-tenant slice data is less accessible under operator confidentiality constraints. The 2026 cohort retains Henna and Rathnayake [27] as its single RW contribution, the remaining 2026 papers being simulation-based, indicating that the most architecturally complex recent work has only begun to reach real-data evaluation.

6.4. O-RAN Interface Compliance Maturity

Interface compliance is assessed at three levels. **Level A** denotes papers that explicitly implement or validate against a named O-RAN interface (E2, O1, A1); **Level B** covers work assuming an O-RAN deployment but relying on simulated or custom interfaces; **Level C**

applies when O-RAN is the stated target but no interface is named or exercised.

Nine papers reach Level A. Panitsas et al. [8] subscribe to E2SM-KPM and dispatch HO commands via E2SM-RC in a closed-loop xApp. Dzaferagic et al. [9] collect measurements via E2SM-KPM v2.0 from a commercial DRAX Near-RT RIC. Groen et al. [3] deploy a CNN xApp on ColO-RAN with a live E2 interface at a 250 ms reporting period. Raftopoulos et al. [23] use E2 for RAN telemetry and A1 for end-to-end latency enrichment on Colosseum (also citing E2SM-CCC). Alchaab et al. [7] deploy on the POWDER testbed using the OSC platform. Kasuluru et al. [17] use the O1 interface bidirectionally for PRB data collection and allocation actuation. Nagib et al. [24] exercise A1 as the communication channel between the Non-RT RIC forecasting module and the Near-RT RIC DRL xApp. Bermudez et al. [30] name O1, A1, E2, and Xn in their deployment workflow. Nguyen et al. [28] explicitly map O1, A1, and E2 to their three-algorithm steps.

Nine papers fall into Level B, referencing O-RAN interface roles without demonstrating compliant message exchange. Ntassah et al. [25] simulate E2 messaging via NATS rather than a standards-compliant stack, while Lotfi and Afghah [6], Asemian et al. [29], Henna and Rathnayake [27], Habib et al. [11], Pérez et al. [18], Azkaei et al. [10], Xiong et al. [19], and Roy et al. [22] name interface roles architecturally without live implementation. The distinction matters for on-wire conformance: Level A works exercise standardized service models (E2SM-KPM, E2SM-RC, and E2SM-CCC) whose message encodings are fixed by the specification, whereas the Level B works and the NATS-based E2 emulation of Ntassah et al. [25] are custom interface surrogates that cannot guarantee interoperability with a standards-compliant E2 node.

Three papers (Rastogi et al. 26, Nassar et al. 20, Nassar et al. 21) are Level C: O-RAN is described conceptually but no interface is specified. Level A papers concentrate in 2024–2025, consistent with growing toolchain maturity (Colosseum, POWDER, OSC), though Level B and Level C papers span the full 2023–2026 range.

6.5. Reproducibility and Open-Source Landscape

Four papers release reproducibility artifacts at varying grades. At the *full* grade, Groen et al. [3] open-source the complete TRACTOR toolchain (capture pipeline, replay engine, CNN xApp) together with an author-collected real-traffic dataset.[1] At the *code-only* grade, Henna and Rathnayake [27] release the Tri-HyperGNN-Sketch training and inference code,[2] evaluated on the external CRAWDAD ElasticMon5G2019[3] dataset rather than an author-released trace. At the *dataset-only* grade, Dzaferagic et al. [9] release a 40,000-sample real O-RAN HO measurement dataset on Mendeley Data. At the *nominal* grade, Nagib et al. [24] reference a code repository,[4] but the link resolves to an empty project as of June 2026.

Testbeds used across the 21 papers include OpenIreland (Accelleran + Benetel + Cumucore), Colosseum (srsRAN SCOPE + ColO-RAN), NSF POWDER (OAI + OSC), the EXata 5G emulator, and custom MATLAB and Python simulators. Datasets span real operator KPM traces (Beijing, European metropolitan 4G), the Huawei SNAP telecom graph, Eurecom ElasticMon5G2019 (CRAWDAD), Telecom Italia Milan cellular traffic, the OSC `ric-app-ad` sample KPI trace, synthetic NWDAF traces, Kumbhar–Shin V2X mobility traces, and Milan O-RAN simulation traces. The absence of a shared benchmark hinders direct metric comparison. The TRACTOR dataset and the OpenIreland HO dataset are the two publicly available O-RAN-native measurement datasets from the surveyed works, and represent the reproducibility floor for future cell-level and UE-level research.

6.6. Temporal Trends (2023–2026)

The 21 papers span four cohorts. The **2023 cohort** ([6, 19, 24–26]; 5 papers) established foundational patterns: traffic forecasting feeding DRL slice managers (Lotfi and Afghah [6], Nagib et al. [24]), cell-level throughput benchmarks (Xiong et al. [19], Ntassah et al. [25]), and classical ML for V2X HO (Rastogi et al. [26]). Interface compliance was sparse: only Nagib et al. [24] genuinely exercised A1 (Level A), whereas Lotfi and Afghah [6] infers A1 from its architecture without a compliant implementation (Level B; section 6.4).

The **2024 cohort** ([3, 11, 18, 20, 22, 23, 28]; 7 papers) shifted toward evaluation rigor: Groen et al. [3] introduced the first O-RAN-compliant KPI dataset from real 5G traffic, Raftopoulos et al. [23] deployed on Colosseum for hardware-in-the-loop evidence, Pérez et al. [18] evaluated on a 3-month production trace with XAI-based behavioral analysis, and Habib et al. [11] intro-

[1] `https://github.com/genesys-neu/TRACTOR`

[2] `https://github.com/shenna2017/ORAN-TS`

[3] `https://ieee-dataport.org/open-access/crawdad-eurecomelasticmon5g2019`

[4] `https://github.com/ahmadnagib/forecasting-aided-DRL`

duced Transformer-family forecasting at TTI granularity inside the Near-RT RIC.

The **2025 cohort** ([7, 9, 10, 17, 21, 30]; 6 papers) is the most diverse: Dzaferagic et al. [9] achieved the survey's only result on a commercial licensed-spectrum O-RAN deployment with E2SM-KPM v2.0, Kasuluru et al. [17] introduced probabilistic forecasting with DYNp as a journal-level rApp contribution, and Bermudez et al. [30] scaled GNN link prediction to operator-scale graphs.

The **2026 cohort** ([8, 27, 29]; 3 papers) marks the current frontier: hypergraph neural networks for multi-connectivity (Henna and Rathnayake [27]), transfer-learning-augmented scalable HO (Panitsas et al. [8]), and hierarchical DRL with Transformer scheduling under adversarial jamming (Asemian et al. [29]). All three are simulation-based, indicating that the most architecturally ambitious contributions remain unvalidated on hardware. The dominant trend across the four cohorts is increasing ML sophistication and growing interface compliance; real-world evaluation remains the primary open gap.

7. Open Challenges and Research Roadmap

Despite encouraging progress across all three traffic granularities, the surveyed literature reveals systematic gaps that constrain real-world deployment. We identify eight open challenges, organized from near-term engineering bottlenecks to longer-horizon research frontiers. Figure 5 summarizes the temporal horizon of each challenge.

7.1. Event-Driven Per-UE xApps

Of the 21 surveyed works, seven operate at the intersection of per-UE granularity and Near-RT RIC placement, distributed across three task families: (i) per-UE resource control: Henna and Rathnayake [27] (hypergraph-based traffic steering) and Nguyen et al. [28] (per-UE beamforming and queue-aware scheduling); (ii) HO prediction: Panitsas et al. [8] and Dzaferagic et al. [9] (per-UE HO classifiers over E2SM-KPM, the latter validated on a commercial O-RAN node); and (iii) per-UE anomaly and safety classification: Azkaei et al. [10] (per-UE KPI anomaly detection within Near-RT RIC time budgets), with conceptual extensions to vehicular and safety-critical scenarios in Asemian et al. [29] and Rastogi et al. [26]. The intersection is therefore inhabited; the open question is no longer whether per-UE Near-RT RIC inference is feasible but how to standardize and harden it for production.

All seven works share a common subscription pattern: per-UE counters delivered via the E2SM-KPM Periodic Report style, which has been the only event trigger style available since v2.0 [33]. Periodic reporting forces every per-UE measurement to wait for the configured reporting interval (from tens of milliseconds to about one second across the surveyed per-UE deployments [8, 9, 27]) before reaching the Near-RT RIC, so a reactive xApp must absorb this latency floor before it can run inference and issue a control action within the same loop period. v7.0 [4], ratified in 2025, addresses this directly by introducing the Measurement Event reporting style, which configures the E2 Node to emit per-UE reports the moment a subscribed measurement crosses an operator-defined condition rather than at fixed intervals.

Because v7.0 is recent and the surveyed works span 2023–2026, none of the seven per-UE Near-RT RIC proposals exploits the Measurement Event style; all rely on Periodic Report subscriptions inherited from v2.0 [5]. The gap is therefore no longer architectural but adoptive: the specification now supports sub-interval per-UE event reporting, but vendor E2 nodes, OSC RIC SDK releases, and xApp reference designs have not yet caught up. Future per-UE xApps targeting reactive control, for example scheduling under bursty load, anticipatory handover, or slice reconfiguration on anomaly detection [5], should adopt Measurement Event subscriptions and report the end-to-end Near-RT RIC loop latency they achieve on real testbeds.

7.2. Real-World Evaluation Scarcity

Cross-cutting analysis of the 21 surveyed papers reveals that 13 rely exclusively on simulation and a further 3 on hardware-in-the-loop emulation testbeds, leaving 5 papers with real-world data. All 5 sit in the RW-trace sub-category: the trained model is fed real operator or network data, but no paper closes the loop with E2SM-RC or A1 policy actuation against a running production deployment during evaluation, and no paper therefore reaches the RW-live sub-category. Dzaferagic et al. [9] collect 40,000 handover samples at a one-second sampling interval on the OpenIreland testbed via an E2SM-KPM v2.0 xApp on the Accelleran DRAX Near-RT RIC, but the LSTM classifier is trained and evaluated offline on a 60-20-20 split with no E2SM-RC actuation closing the loop. Henna and Rathnayake [27] use the Eurecom ElasticMon5G2019 dataset collected on an OpenAirInterface RAN and core deployment, evaluating link prediction offline on an 80/20 split. Xiong et al. [19] use 28 days of KPM data from 48 production base stations near Beijing; Pérez et al. [18] use a three-month

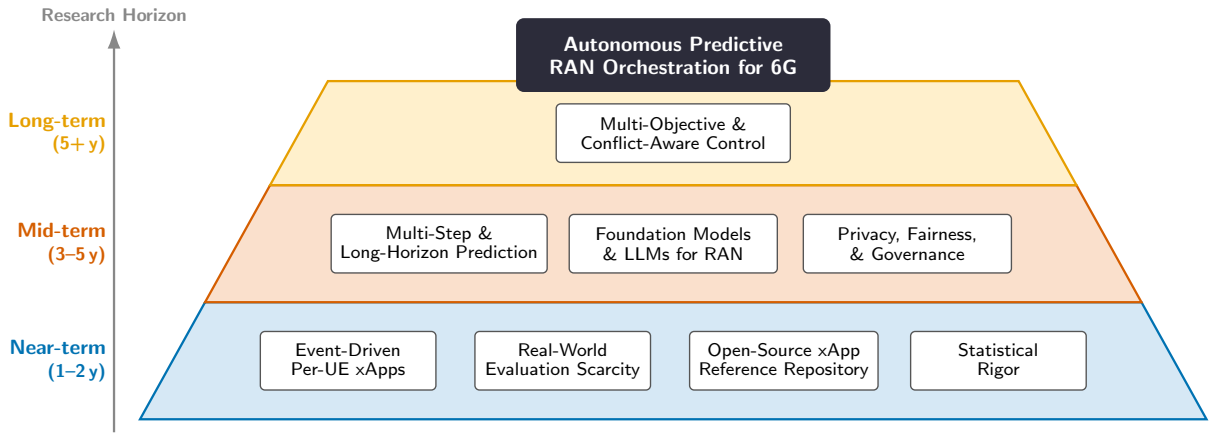

Figure 5: Research roadmap for learning-based traffic prediction in O-RAN, organized as a convergence pyramid. The near-term band (1–2 years) identifies four engineering bottlenecks: building event-driven per-UE xApps on E2SM-KPM Measurement Event subscriptions, expanding real-world evaluation, building an open-source xApp reference repository, and enforcing statistical rigor. The mid-term band (3–5 years) targets three emerging research frontiers: multi-step and long-horizon prediction, foundation models and LLMs for RAN, and privacy, fairness, and governance. The long-term band (5+ years) converges on multi-objective and conflict-aware control, culminating in fully autonomous predictive RAN orchestration for 6G.

4G production trace from a European metropolitan area; and Bermudez et al. [30] use an anonymized Huawei operator graph at million-UE scale. The absence of any RW-live work is therefore the primary structural gap and the most concrete open-challenge target for the next research cycle. Simulation results are useful for controlled ablation, but they routinely overestimate generalization: idealized channel models suppress the bursty interference patterns that characterize dense urban deployments, synthetic traffic generators cannot replicate diurnal load anomalies, and emulated E2 interfaces omit the vendor-specific timing jitter and measurement gaps introduced by real HO events.

Access to production O-RAN testbeds remains a fundamental barrier to reproducible experimentation: POWDER, Arena, and Colosseum provide shared access, but each suffers from long queue times, heterogeneous node configurations that change between allocations, and the absence of standardized traffic scenarios and reporting windows that would permit cross-paper comparison. Vendor-hosted closed testbeds are rarely accessible to academic researchers, compounding the problem. The community needs a standardized O-RAN benchmark protocol analogous to MLPerf [34]: a defined set of traffic scenarios (bursty eMBB, latency-sensitive URLLC, and mixed-slice), standardized traffic datasets with fixed train and test splits so that every method is scored on identical data, agreed KPI sets (MAE, RMSE, and 95th-percentile error on a common holdout window), and mandatory disclosure of E2 report periodicity and RIC latency budgets. A curated, versioned repository of open RAN traces, extending the footprint established by Groen et al. [3], would substantially lower the barrier to reproducible, realistic evaluation.

A further deployment barrier is multi-vendor scalability: O-RAN's central value proposition is interoperability across equipment from different vendors [1], yet a predictor trained on one vendor's KPM semantics and RU behavior need not transfer to another, and a single Near-RT RIC may have to host and scale prediction xApps across many vendors' radio units and distributed units simultaneously. The per-environment transfer learning of Panitsas et al. [8] mitigates the retraining cost of topology changes, but cross-vendor portability of trained models, alongside the conflict mitigation that arises when xApps from multiple vendors contend for the same control surface (section 7.8), remains an open deployment-scalability concern.

7.3. Open-Source xApp Reference Repository

Reproducibility in O-RAN prediction research is critically limited by the near-total absence of openly released implementations. Among all 21 surveyed works, only Groen et al. [3] satisfy the minimum bar for reproductive completeness: source code, a real-traffic (non-synthetic) dataset, and training/evaluation scripts are all publicly available through the TRACTOR project. Every other paper describes its approach at the algorithm level only, leaving a researcher who wishes to replicate

the work to re-implement from scratch: E2 subscription logic and E2SM message parsing, RIC SDK integration against the OSC Near-RT RIC, time-series preprocessing pipelines specific to each RAN metric, and end-to-end training and evaluation harnesses.

The field needs a community reference repository with four components: (i) a `data/` directory hosting open RAN trace collections with a standardized schema (columns: `timestamp`, `cell_id`, `ue_id`, `metric_name`, `value`), analogous to Hugging Face Datasets for natural language processing; (ii) a `models/` directory of framework-agnostic model definitions in PyTorch and ONNX, analogous to the ONNX Model Zoo; (iii) an `xapp/` directory providing a minimal xApp harness compliant with the OSC Near-RT RIC SDK and the E2SM-KPM service model; and (iv) a `benchmarks/` directory with evaluation scripts that reproduce key results. Without such infrastructure, the MAE and RMSE figures reported across the 21 surveyed papers are incommensurable, and the gap between a published algorithm and a deployable xApp remains wide.

7.4. Statistical Rigor

A recurring methodological weakness across the surveyed literature is that most reported MAE, RMSE, PRB allocation accuracy, and SLA compliance rate figures derive from single experimental runs, with variance information rarely reported. Kasuluru et al. [17] is a partial exception, providing structured multi-model comparisons with numerical results across probabilistic forecasters; yet even that work reports no multi-seed standard deviations. Without variance reporting, it is impossible to determine whether a published gain reflects a genuine algorithmic advantage or an artifact of initialization luck or dataset specificity: a model trained on one diurnal traffic profile may fail to generalize to a shifted hour distribution encountered in a different cell or deployment context.

Auditing the corpus against seven established rigor practices confirms how thinly they are applied. Cross-validation appears in 2 of the 21 works: Panitsas et al. [8] runs 10-fold cross-validation, and Nassar et al. [21] folds it into a GridSearch procedure; the rest use a single fixed split. Multi-seed reporting with standard deviation appears in 1 work, Henna and Rathnayake [27], which trains across three random seeds and reports mean ± standard deviation; no other work quantifies run-to-run variance. Ablation studies appear in 2 works, Henna and Rathnayake [27] and Asemian et al. [29], which disable individual architectural components and report the effect. Systematic hyperparameter tuning is the most common practice, present in 4 works (Nassar et al. 21, Azkaei et al. 10, Ntassah et al. 25, Panitsas et al. 8), typically grid search over a small predefined space. Regularization (dropout or batch normalization) is mentioned in a handful of deep-learning works but is rarely tied to a reported effect on generalization. Uncertainty quantification appears in 1 work, Kasuluru et al. [17], via DeepAR and quantile-based probabilistic forecasting. Explicit drift or noise-robustness testing appears in 1 work, Nagib et al. [24], which sweeps Gaussian forecast-error perturbations and reports the convergence breakdown point. No work in the corpus reports bootstrap confidence intervals on downstream control gains, and none benchmarks against a persistence or autoregressive moving-average baseline.

The remedy is to import the reporting norms already established in the broader time-series forecasting community, not to invent new ones. The Monash Forecasting Benchmark, the M-competition series, and NeurIPS time-series tracks all mandate: (i) results averaged over at least five independent random seeds with reported standard deviation; (ii) ablation tables disabling one architectural component at a time; (iii) bootstrap confidence intervals on downstream control gains; and (iv) explicit comparison against a persistence baseline and an autoregressive moving-average benchmark. Journals reviewing O-RAN prediction submissions should treat the absence of multi-seed evaluation as grounds for revision.

7.5. Multi-Step and Long-Horizon Prediction

Most surveyed models produce a single-step forecast: a point estimate of traffic at $t + 1$, consumed immediately by the xApp within the Near-RT RIC's 10 ms–1 s control loop. This timescale is too short for the Non-RT RIC, whose rApps plan policies over horizons of 10 s–60 s and require $T = 10$–30 look-ahead steps to solve a meaningful optimization. The gap means single-step xApp outputs cannot directly feed rApp planning; a dedicated multi-step predictor operating at the rApp timescale is needed. Pérez et al. [18] address this directly by evaluating LSTM, DLinear, and PatchTST forecasters at horizons up to 30 steps (and 60 steps on D1) on O-RAN traces. In addition, Kasuluru et al. [17] further supply calibrated uncertainty via DeepAR and quantile regression, establishing that multi-step probabilistic forecasting is technically feasible in O-RAN deployments. Outside the O-RAN context, lightweight temporal attention hybrids have also demonstrated competitive accuracy for 5G traffic forecasting [35].

The natural architecture for consuming such forecasts is an rApp based on model predictive control (MPC): at

each planning cycle, the rApp ingests the T-step predictive distribution $\hat{p}(x_{t+1:t+T})$, solves a constrained optimization over PRB allocation subject to SLA constraints and HO cost, and commits the resulting policy π_t for the next T steps via an A1-P message to the Near-RT RIC. Point-forecast MPC breaks down when actual traffic exceeds the predicted mean: SLA violations accumulate undetected until the next re-planning cycle, and the rApp has no signal to trigger early re-optimization. Probabilistic forecasts from models such as DeepAR [17] expose this uncertainty explicitly, enabling chance-constrained formulations that trade off over-provisioning against violation risk; coupling such outputs to the Non-RT RIC optimization loop remains a high-impact open direction.

7.6. Foundation Models and LLMs for RAN

Evidence from the broader ML community suggests that pre-trained language model representations transfer surprisingly well to temporal sequences: Zhou et al. [36] show that frozen GPT-2 self-attention and feed-forward blocks serve as a universal time-series backbone across six analysis tasks (classification, forecasting, imputation, anomaly detection, few-shot and zero-shot learning), with only the embedding, normalization, and output layers fine-tuned per task, and Jin et al. [37] achieve state-of-the-art accuracy on standard benchmarks by reprogramming an LLM's input space with time-series patch tokens, particularly in few-shot and zero-shot settings. Liang et al. [38] survey this maturing field, cataloguing pre-training objectives and adaptation strategies for time-series foundation models. Narrowing to RAN, Panitsas et al. [8] demonstrate that task-specific transfer already delivers value: a HO predictor pre-trained on one radio environment substantially outperforms a model trained from scratch when fine-tuned on a new environment. A true RAN foundation model, however, would require self-supervised pre-training on diverse multi-vendor, multi-frequency, multi-scenario cellular traces that jointly capture temporal and spatial wireless correlations.

The telecom specialization work of Zou et al. [39] points toward a second opportunity: TelecomGPT, trained through continual pre-training, instruction tuning, and alignment on curated telecom corpora, outperforms GPT-4 on telecom math modeling and code generation tasks, demonstrating that LLMs can acquire domain-specific reasoning. This foundation makes intent-driven rApp policy generation plausible: an LLM-based rApp could translate natural-language SLA specifications into A1 policy configurations or O1 management intents. Three open research questions remain: how to ground LLM outputs in the formal semantics of E2SM and A1-P schemas; how to verify that generated policies are feasible and safe before deployment; and how to quantify hallucination risk in safety-critical control paths.

7.7. Privacy, Fairness, and Governance

Traffic prediction in multi-vendor O-RAN deployments generates telemetry that is commercially sensitive: per-UE throughput profiles expose application usage patterns, and slice-level demand forecasts reveal operator capacity planning assumptions. Training a joint prediction model spanning RUs from multiple vendors requires sharing such data across organizational boundaries, making federated learning (FL) a natural fit. Lim et al. [40] and Niknam et al. [41] establish FL's applicability to mobile edge and wireless settings, showing that local model updates can be aggregated without raw data exchange; complementary work by Kwon et al. [42] develops multi-agent DDPG with federated learning for networked IoT resource allocation, illustrating that FL can coexist with DRL-based wireless control. Asynchronous FL with DRL-based caching has similarly reduced content access latency in vehicular edge networks [43]. Roy et al. [22] instantiate this approach in O-RAN, demonstrating federated machine reasoning for resource provisioning with accuracy comparable to centralized training while preserving data locality.

Differential privacy (DP) strengthens FL by adding a formal (ε, δ)-guarantee to gradient updates [44], but at the cost of reduced model accuracy; the optimal DP budget for O-RAN telemetry remains unstudied. Non-IID data distributions across participants degrade FedAvg convergence, as surveyed in Lim et al. [40], and transmitting model updates introduces communication overhead in federated learning over wireless channels [41]. Lightweight FL variants using data pruning can mitigate resource constraints while preserving accuracy, as shown for IIoT intrusion detection [45]. A further tension arises from auditability: regulators may require that prediction models driving resource allocation be explainable, which conflicts with the opacity of large models trained under DP constraints. Systematic study of the privacy-utility-communication trade-off, coupled with standardized Non-RT RIC interfaces for privacy-preserving model aggregation, constitutes a high-priority research direction.

Beyond privacy, the forecast-then-act loop raises an operational-risk concern that the surveyed corpus partially quantifies: the controller acts on predictions that are sometimes wrong. Nagib et al. [24] model forecast error as zero-mean Gaussian and report that their

forecasting-aided DRL converges reliably only up to an error standard deviation of about 0.25 relative to a unit forecast range, above which the agents fail to converge, whereas acting on the forecast alone never converges. Habib et al. [11] mitigate this exposure by gating E2 actuation on load thresholds, so the controller does not act on every prediction, and Kasuluru et al. [17] argue that point forecasts force blind decisions and instead use probabilistic forecasting, tuning a percentile to balance the asymmetric costs of over- and under-provisioning. Fairness across slices and users is, by contrast, largely unaddressed as an explicit objective. Only Lotfi and Afghah [6] reports distributional fairness evidence, through per-slice QoS-violation spread and per-user throughput distributions, and even there fairness is observed rather than optimized, while Raftopoulos et al. [23] treat slices independently and explicitly defer cross-slice coordination to future work. A predictor that drives allocation toward an aggregate reward can therefore starve low-priority slices or users, a risk the corpus does not measure.

Governance of forecast-driven control rests on transparency, accountability, and bias control, each only partially addressed. For transparency, Pérez et al. [18] apply SHAP-based XAI and observe that the most accurate forecasters are not self-explainable, which can disqualify them from production in favor of interpretable models, whereas Roy et al. [22] build interpretability in through a neuro-symbolic LTN design; this explainability requirement compounds the opacity-versus-DP tension noted above. For accountability, the O-RAN AI/ML lifecycle already provides governance substrate, since model-performance monitoring reported over O1 can trigger automatic model termination on degradation [16], and O1 trace and performance-management services supply the supporting telemetry [32], yet accountability for individual forecast-driven control decisions is not yet specified. For bias, the training data is narrow and simulation-dominated, with the few real traces drawn from single operators and single regions, such as one district near Beijing in 2018 [19] and one European operator [18], and no surveyed work analyzes dataset bias or cross-operator and cross-region generalization.

7.8. Multi-Objective and Conflict-Aware Control

A single O-RAN deployment may host dozens of xApps and rApps simultaneously, each optimizing a distinct objective: throughput maximization, latency minimization, energy efficiency, interference management, and SLA compliance. The O-RAN specifications define three conflict types [46]: *direct* conflicts (two xApps issuing back-to-back control messages that modify the same RAN parameter, so only the last takes effect); *indirect* conflicts (xApps modifying different parameters that influence the same RAN area, producing oscillatory behavior); and *implicit* conflicts (xApps pursuing separate goals over disjoint parameter sets whose joint influence degrades network KPIs in a non-obvious manner). Adamczyk and Kliks [47] build on these definitions to propose a conflict mitigation framework (CMF) for the Near-RT RIC, formalizing detection and resolution procedures for each type. In Bonati et al. [48], twelve DRL agents run in parallel on the Near-RT RIC, each managing scheduling policy and resource allocation for a distinct slice with independently trained reward functions, exemplifying how independent optimization objectives create potential for resource contention. Outside the O-RAN setting, cooperative multi-agent DRL frameworks such as the CommNet-based scheduler of Park et al. [49] show that explicit inter-agent communication can align competing objectives, pointing to a design pattern that conflict-aware xApp coordination layers may eventually adopt. Habib et al. [11] address a narrow instance of this problem through threshold-gated actuation: a Transformer predictor suppresses E2 control messages unless the load forecast exceeds a configurable threshold, avoiding unnecessary single-xApp interventions but offering no mechanism for coordinating competing xApps.

A principled solution requires formalizing xApp interactions as a multi-objective optimization problem: let each xApp i define a reward function r_i and an E2 action constraint set $\mathcal{A}_i$; a conflict resolution layer must find a joint action $a^* \in \bigcap_i \mathcal{A}_i$ on or near the Pareto frontier of $\{r_i\}$. D'Oro et al. [50] prove that this orchestration problem is NP-hard in general and propose low-complexity heuristics executed in the Non-RT RIC that prevent data-driven algorithms from issuing conflicting commands over the same parameter sets, validated on Colosseum with 7 base stations and 42 UEs. Despite this progress, O-RAN has standardized the conflict-mitigation taxonomy and a study-level framework [46, 51], but normative mechanisms for detecting and resolving all xApp conflict scenarios remain incomplete, leaving detailed procedure standardization open. Extending the A1-P schema to carry machine-readable objective declarations and deploying a Non-RT RIC conflict resolution engine that computes Pareto-optimal joint policies would transform the O-RAN application layer from a set of independently optimizing controllers into a coordinated multi-objective management system, an architectural prerequisite for the controller densities anticipated in 6G deployments.

8. Conclusion

This survey examined 21 technical works on ML-based traffic and UE behavior prediction in O-RAN, classifying each along four axes: traffic target granularity, ML tier, evaluation environment, and O-RAN interface placement. Analysis across the full corpus reveals a field that has converged on a dominant architectural pattern but that systematically overstates real-world readiness. The strongest evidence is the forecast-then-act architectural convergence, demonstrated across fourteen papers and all three prediction granularities; the weakest concerns real-world readiness, reproducibility, and statistical rigor, where no surveyed work performs live closed-loop actuation, only one releases a complete reproducibility package, and few report multi-seed variance.
The principal findings are as follows.

- **The forecast-then-act pipeline dominates.** Fourteen of the 21 papers compose a predictor with a downstream controller, feeding forecasts into DRL state vectors (V1), threshold-based application gating (V2), or direct rule-based action selection (V3). This pipeline is the de facto closed-loop design across all three granularities.

- **Standard deep learning (Tier 2) predominates over novel architectures.** Eleven papers apply established deep learning methods to O-RAN contexts, while only five employ Tier 1 architectures with structural inductive bias tailored to RAN prediction tasks. Classical and tabular approaches account for the remaining five.

- **Evaluation fidelity is low.** Sixteen of the 21 papers evaluate only in simulation or hardware-in-the-loop emulation; five evaluate on real-world operator traces, and none perform live closed-loop actuation on a production O-RAN deployment. No two of the real-data papers share a common traffic scenario, reporting window, or KPI set, making cross-paper comparison infeasible.

- **Per-UE xApp adoption lags the standard.** Seven of 21 surveyed works place per-UE inference at the Near-RT RIC, yet all rely on E2SM-KPM Periodic Report subscriptions; none have adopted the Measurement Event style added in v7.0 [5], leaving sub-interval reactive control as the chief open frontier rather than the existence of per-UE xApps as previously framed.

- **Reproducibility is near-absent.** Among all 21 surveyed works, only one provides a unified release of source code, real-world traffic data, and evaluation scripts [3]. The remaining studies lack a comprehensive reproducibility package combining these essential artifacts.

- **Statistical rigor is weak.** The large majority of reported accuracy figures derive from single experimental runs; only a few works report multi-seed variance, cross-validation, or ablation, and none provide bootstrap confidence intervals or a persistence baseline, making it hard to distinguish genuine algorithmic gains from initialization artifacts.

These gaps define the near-term research agenda [5]: per-UE Near-RT RIC xApps that exploit the E2SM-KPM v7.0 Measurement Event style for sub-interval reactive control, a standardized E2SM prediction profile exposing per-UE counters at configurable granularity, a community reference repository of open RAN traces and xApp harnesses, and mandatory multi-seed evaluation norms for journal submissions in this area.

Looking further ahead, the convergence of O-RAN's open interfaces with foundation models pre-trained on large-scale RAN corpora, privacy-preserving federated prediction across multi-operator deployments, and multi-objective conflict-aware control policies will define the trajectory toward 6G predictive intelligence. Realizing this trajectory requires the community to first close the validation gap: algorithms that cannot be reproduced, evaluated on real traffic, or benchmarked against a common baseline do not provide a reliable foundation for the next architectural generation.

Appendix: List of Abbreviations

This appendix lists the abbreviations used throughout this paper, in alphabetical order, for quick reference.

3GPP	3rd Generation Partnership Project
A1-EI	A1 Enrichment Information
A1-P	A1 Policy Management
AI	Artificial Intelligence
CNN	Convolutional Neural Network
CP	Cyclic Prefix
CPU	Central Processing Unit
CQI	Channel Quality Indicator
dApp	Distributed Application
DDPG	Deep Deterministic Policy Gradient
DDQN	Double Deep Q-Network
DL	Downlink

DP	Differential Privacy
DQN	Deep Q-Network
DRL	Deep Reinforcement Learning
E2AP	E2 Application Protocol
E2SM	E2 Service Model
E2SM-CCC	E2SM Cell Configuration and Control
E2SM-KPM	E2SM Key Performance Measurement
E2SM-RC	E2SM RAN Control
eCPRI	Enhanced Common Public Radio Interface
eMBB	Enhanced Mobile Broadband
FCAPS	Fault, Configuration, Accounting, Performance, and Security
FFT	Fast Fourier Transform
FL	Federated Learning
GAT	Graph Attention Network
GCN	Graph Convolutional Network
gNB	Next-Generation NodeB
GNB	Gaussian Naive Bayes
GNN	Graph Neural Network
GRU	Gated Recurrent Unit
HARQ	Hybrid Automatic Repeat Request
HO	Handover
HTTPS	Hypertext Transfer Protocol Secure
IFFT	Inverse Fast Fourier Transform
IIoT	Industrial Internet of Things
KNN	k-Nearest Neighbors
KPI	Key Performance Indicator
KPM	Key Performance Measurement
LLM	Large Language Model
LSTM	Long Short-Term Memory
LTN	Logic Tensor Network
MAC	Medium Access Control
MAE	Mean Absolute Error
MCS	Modulation and Coding Scheme
ML	Machine Learning
mMTC	Massive Machine-Type Communications
MPC	Model Predictive Control
MSE	Mean Squared Error
Near-RT RIC	Near-Real-Time RIC
NETCONF	Network Configuration Protocol
Non-RT RIC	Non-Real-Time RIC
NSSI	Network Slice Subnet Instance
NWDAF	Network Data Analytics Function
O-CU	O-Central Unit
O-CU-CP	O-CU Control Plane
O-CU-UP	O-CU User Plane
O-DU	O-Distributed Unit
OFDMA	Orthogonal Frequency-Division Multiple Access
ONNX	Open Neural Network Exchange
O-RAN	Open Radio Access Network
O-RU	O-Radio Unit
OSC	O-RAN Software Community
PCI	Physical Cell Identity
PDCP	Packet Data Convergence Protocol
PLFS	Physical Layer Frequency Signals
PPO	Proximal Policy Optimization
PRB	Physical Resource Block
PTP	Precision Time Protocol
QoS	Quality of Service
RAN	Radio Access Network
rApp	Non-RT RIC Application
RB	Resource Block
REST	Representational State Transfer
RF	Radio Frequency
RIC	RAN Intelligent Controller
RL	Reinforcement Learning
RLC	Radio Link Control
RMSE	Root Mean Squared Error
RNN	Recurrent Neural Network
RRC	Radio Resource Control
RSRP	Reference Signal Received Power
RSRQ	Reference Signal Received Quality
RSSINR	Received Signal Strength Indicator-to-Interference-plus-Noise Ratio
RUT	Resource Utilization Ratio
SAC	Soft Actor-Critic
SDAP	Service Data Adaptation Protocol
SDK	Software Development Kit
SDL	Shared Data Layer
SHAP	Shapley Additive Explanations
SINR	Signal-to-Interference-plus-Noise Ratio
SLA	Service Level Agreement
SMO	Service Management and Orchestration
SSH	Secure Shell
TLS	Transport Layer Security
TTI	Transmission Time Interval
UE	User Equipment
UEID	UE Identifier

URLLC	Ultra-Reliable Low-Latency Communications
V2X	Vehicle-to-Everything
vBS	Virtualized Base Station
VGAE	Variational Graph Autoencoder
WG	Working Group
XAI	Explainable Artificial Intelligence
xApp	Near-RT RIC Application
YANG	Yet Another Next Generation

References

[1] M. Polese, L. Bonati, S. D'Oro, S. Basagni, T. Melodia, Understanding O-RAN: Architecture, Interfaces, Algorithms, Security, and Research Challenges, IEEE Communications Surveys & Tutorials 25 (2023) 1376–1411.

[2] K. Alam, M. A. Habibi, M. Tammen, D. Krummacker, W. Saad, M. D. Renzo, T. Melodia, X. Costa-Pérez, M. Debbah, A. Dutta, H. D. Schotten, A Comprehensive Tutorial and Survey of O-RAN: Exploring Slicing-Aware Architecture, Deployment Options, Use Cases, and Challenges, IEEE Communications Surveys & Tutorials 28 (2026) 1637–1678.

[3] J. Groen, M. Belgiovine, U. Demir, B. Kim, K. Chowdhury, TRACTOR: Traffic Analysis and Classification Tool for Open RAN, in: ICC 2024 - IEEE International Conference on Communications, pp. 4894–4899. ISSN: 1938-1883.

[4] O-RAN Working Group 3, O-RAN Near-Real-time RAN Intelligent Controller E2 Service Model (E2SM), KPM 7.0, Technical Specification O-RAN.WG3.TS.E2SM-KPM-R004-v07.00, O-RAN Alliance, Alfter, Germany, 2025.

[5] T. T.-A. Dang, D. Won, J. Kim, V. A.-H. Le, J. Kim, S. Cho, Toward Event-Driven Per-UE xApps in Open RAN Near-RT RIC, in: 2026 Seventeenth International Conference on Ubiquitous and Future Networks (ICUFN), IEEE, 2026. To appear.

[6] F. Lotfi, F. Afghah, Open RAN LSTM Traffic Prediction and Slice Management Using Deep Reinforcement Learning, in: 2023 57th Asilomar Conference on Signals, Systems, and Computers, pp. 646–650. ISSN: 2576-2303.

[7] A. Alchaab, A. Younis, D. Pompili, Slice-on-the-Fly: AI-based Network Slicing in O-RAN for Dynamic Traffic Demands, in: 2025 IEEE 26th International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM), IEEE, Fort Worth, TX, USA, 2025, pp. 51–60.

[8] I. Panitsas, A. Mudvari, A. Maatouk, L. Tassiulas, Predictive Handover Strategy in 6G and Beyond: A Deep and Transfer Learning Approach, 2026. ArXiv:2404.08113 [cs].

[9] M. Dzaferagic, B. Missi Xavier, D. Collins, V. D'Onofrio, M. Martinello, M. Ruffini, ML-Based Handover Prediction Over a Real O-RAN Deployment Using RAN Intelligent Controller, IEEE Transactions on Network and Service Management 22 (2025) 635–647.

[10] B. Azkaei, K. C. Joshi, G. Exarchakos, Machine Learning-Driven Anomaly Detection for 5G O-RAN Performance Metrics, in: IEEE INFOCOM 2025 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), pp. 1–6. ISSN: 2833-0587.

[11] M. A. Habib, P. E. I. Rivera, Y. Ozcan, M. Elsayed, M. Bavand, R. Gaigalas, M. Erol-Kantarci, Transformer-Based Wireless Traffic Prediction and Network Optimization in O-RAN, in: 2024 IEEE International Conference on Communications Workshops (ICC Workshops), pp. 1–6.

[12] A. Elyasi, A. Ashdown, K. M. Rumman, F. Restuccia, O-Ran Xapps: Survey and Research Challenges, 2025.

[13] N. Mthethwa, J. Mwangama, M. Masonta, Review of Machine Learning Techniques that Enable Network Slicing in Open RAN, in: 2025 IEEE 3rd Wireless Africa Conference (WAC), pp. 1–6.

[14] 3GPP, NG-RAN; Architecture description, Technical Specification 3GPP TS 38.401, 3rd Generation Partnership Project (3GPP), 2026. Version 19.2.0, Release 19.

[15] O-RAN Working Group 1, O-RAN Architecture Description 16.0, Technical Report O-RAN.WG1.TS.OAD-R005-v16.00, O-RAN Alliance, Alfter, Germany, 2026.

[16] O-RAN Working Group 2, O-RAN AI/ML workflow description and requirements 1.03, Technical Report O-RAN.WG2.AIML-v01.03, O-RAN Alliance, Alfter, Germany, 2021.

[17] V. Kasuluru, L. Blanco, E. Zeydan, Enhancing Open RAN Operations: The Role of Probabilistic Forecasting in Network Analysis, IEEE Transactions on Network and Service Management 22 (2025) 3676–3691.

[18] P. F. Pérez, C. Fiandrino, M. Fiore, J. Widmer, An In-Depth Analysis of Advanced Time Series Forecasting Models for the Open RAN, in: IEEE INFOCOM 2024 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), pp. 1–6. ISSN: 2833-0587.

[19] Z. Xiong, K. Zhang, G. Chuai, X. Yang, Y. Xu, Intelligent Cellular Traffic Prediction in Open-RAN Based on Cross-Domain Data Fusion, in: IEEE INFOCOM 2023 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), pp. 1–6. ISSN: 2833-0587.

[20] A. W. Nassar, H. ALy, H. M. ElBadawy, Cell Throughput Prediction Using AI Models: Insights from the O-RAN Framework, in: 2024 6th Novel Intelligent and Leading Emerging Sciences Conference (NILES), pp. 589–592.

[21] A. W. Nassar, H. Aly, H. M. ElBadawy, Machine Learning Cell Throughput Prediction to Enhance Handover Processes: A Hyperparameter-Tuned Approach within the ORAN Framework, in: 2025 42nd National Radio Science Conference (NRSC), volume 1, pp. 118–126. ISSN: 2837-018X.

[22] S. Roy, H. Chergui, A. Ksentini, C. Verikoukis, Federated Machine Reasoning for Resource Provisioning in 6G O-RAN, 2024. ArXiv:2406.06128 [cs].

[23] R. Raftopoulos, S. D'Oro, T. Melodia, G. Schembra, DRL-based Latency-Aware Network Slicing in O-RAN with Time-Varying SLAs, in: 2024 International Conference on Computing, Networking and Communications (ICNC), pp. 737–743. ArXiv:2401.05042 [cs].

[24] A. M. Nagib, H. Abou-zeid, H. S. Hassanein, How Does Forecasting Affect the Convergence of DRL Techniques in O-RAN Slicing?, in: GLOBECOM 2023 - 2023 IEEE Global Communications Conference, pp. 2644–2649. ISSN: 2576-6813.

[25] R. Ntassah, G. M. Dell'Aera, F. Granelli, xApp for Traffic Steering and Load Balancing in the O-RAN Architecture, in: ICC 2023 - IEEE International Conference on Communications, pp. 5259–5264. ISSN: 1938-1883.

[26] E. Rastogi, M. K. Maheshwari, J. P. Jeong, Intelligent O-RAN-Based Proactive Handover in Vehicular Networks, in: 2023 14th International Conference on Information and Communication Technology Convergence (ICTC), pp. 481–486. ISSN: 2162-1241.

[27] S. Henna, U. Rathnayake, Hypergraph Representation Learning-Based xApp for Traffic Steering in 6G O-RAN Closed-Loop Control, IEEE Transactions on Network and Service Management 23 (2026) 2190–2209.

[28] V.-D. Nguyen, T. X. Vu, N. T. Nguyen, D. C. Nguyen, M. Juntti,

N. C. Luong, D. T. Hoang, D. N. Nguyen, S. Chatzinotas, Network-Aided Intelligent Traffic Steering in 6G O-RAN: A Multi-Layer Optimization Framework, IEEE Journal on Selected Areas in Communications 42 (2024) 389–405.

[29] G. Asemian, M. Amini, B. Kantarci, Anti-Jamming Task Scheduling in MEC-O-RAN With Hierarchical DRL and Transformer-Based Control, IEEE Internet of Things Journal 13 (2026) 7714–7729.

[30] A. G. Bermudez, M. Farreras, M. Groshev, J. A. Trujillo, I. d. l. Bandera, R. Barco, Graph Neural Networks for O-RAN Mobility Management: A Link Prediction Approach, 2025. ArXiv:2502.02170 [cs].

[31] G. Barlacchi, M. De Nadai, R. Larcher, A. Casella, C. Chitic, G. Torrisi, F. Antonelli, A. Vespignani, A. Pentland, B. Lepri, A multi-source dataset of urban life in the city of Milan and the Province of Trentino, Scientific Data 2 (2015) 150055.

[32] O-RAN Working Group 10, O-RAN O1 Interface Specification 18.0, Technical Specification O-RAN.WG10.TS.O1-Interface.0-R005-v18.00, O-RAN Alliance, Alfter, Germany, 2026.

[33] O-RAN Working Group 3, O-RAN Near-Real-time RAN Intelligent Controller E2 Service Model (E2SM), KPM 2.0, Technical Specification O-RAN.WG3.E2SM-KPM-v02.00, O-RAN Alliance, Alfter, Germany, 2021.

[34] P. Mattson, C. Cheng, G. Diamos, C. Coleman, P. Micikevicius, D. Patterson, H. Tang, G.-Y. Wei, P. Bailis, V. Bittorf, D. Brooks, D. Chen, D. Dutta, U. Gupta, K. Hazelwood, A. Hock, X. Huang, D. Kang, D. Kanter, N. Kumar, J. Liao, D. Narayanan, T. Oguntebi, G. Pekhimenko, L. Pentecost, V. Janapa Reddi, T. Robie, T. St John, C.-J. Wu, L. Xu, C. Young, M. Zaharia, MLPerf Training Benchmark, Proceedings of Machine Learning and Systems 2 (2020) 336–349.

[35] D. S. Samudrala, R. Senapati, Advanced temporal attention mechanism based traffic prediction model for 5G and beyond cellular networks, ICT Express (2025).

[36] T. Zhou, P. Niu, X. Wang, L. Sun, R. Jin, One Fits All: Power General Time Series Analysis by Pretrained LM, Advances in Neural Information Processing Systems 36 (2023) 43322–43355.

[37] M. Jin, S. Wang, L. Ma, Z. Chu, J. Y. Zhang, X. Shi, P.-Y. Chen, Y. Liang, Y.-F. Li, S. Pan, others, Time-llm: Time series forecasting by reprogramming large language models, arXiv preprint arXiv:2310.01728 (2023).

[38] Y. Liang, H. Wen, Y. Nie, Y. Jiang, M. Jin, D. Song, S. Pan, Q. Wen, Foundation Models for Time Series Analysis: A Tutorial and Survey, in: Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD '24, Association for Computing Machinery, New York, NY, USA, 2024, pp. 6555–6565.

[39] H. Zou, Q. Zhao, Y. Tian, L. Bariah, F. Bader, T. Lestable, M. Debbah, TelecomGPT: A Framework to Build Telecom-Specific Large Language Models, IEEE Transactions on Machine Learning in Communications and Networking 3 (2025) 948–975.

[40] W. Y. B. Lim, N. C. Luong, D. T. Hoang, Y. Jiao, Y.-C. Liang, Q. Yang, D. Niyato, C. Miao, Federated Learning in Mobile Edge Networks: A Comprehensive Survey, IEEE Communications Surveys & Tutorials 22 (2020) 2031–2063.

[41] S. Niknam, H. S. Dhillon, J. H. Reed, Federated Learning for Wireless Communications: Motivation, Opportunities, and Challenges, IEEE Communications Magazine 58 (2020) 46–51.

[42] D. Kwon, J. Jeon, S. Park, J. Kim, S. Cho, Multiagent DDPG-Based Deep Learning for Smart Ocean Federated Learning IoT Networks, IEEE Internet of Things Journal 7 (2020) 9895–9903.

[43] A. K. Sinthia, N. I. Mahbub, E.-N. Huh, Optimizing proactive content caching with mobility aware deep reinforcement & asynchronous federate learning in VEC, ICT Express 11 (2025) 293–298.

[44] C. Dwork, A. Roth, The Algorithmic Foundations of Differential Privacy, Foundations and Trends in Theoretical Computer Science 9 (2014) 211–487.

[45] S.-J. Lee, I.-G. Lee, Lightweight federated learning-based intrusion detection system for industrial internet of things, ICT Express 11 (2025) 690–695.

[46] O-RAN Working Group 3, O-RAN Conflict Mitigation 1.0, Technical Report O-RAN.WG3.TR.ConMit-R004-v01.00, O-RAN Alliance, Alfter, Germany, 2024.

[47] C. Adamczyk, A. Kliks, Conflict mitigation framework and conflict detection in O-RAN near-RT RIC, IEEE Communications Magazine 61 (2023) 199–205.

[48] L. Bonati, S. D'Oro, M. Polese, S. Basagni, T. Melodia, Intelligence and Learning in O-RAN for Data-Driven NextG Cellular Networks, IEEE Communications Magazine 59 (2021) 21–27.

[49] S. Park, C. Park, J. Kim, Learning-Based Cooperative Mobility Control for Autonomous Drone-Delivery, IEEE Transactions on Vehicular Technology 73 (2024) 4870–4885.

[50] S. D'Oro, L. Bonati, M. Polese, T. Melodia, OrchestRAN: Network Automation through Orchestrated Intelligence in the Open RAN, in: IEEE INFOCOM 2022 - IEEE Conference on Computer Communications, pp. 270–279.

[51] O-RAN Working Group 2, O-RAN Study on A1 policy conflict mitigation 1.0, Technical Report O-RAN.WG2.TR.PlcCnfMtgt-R004-v01.00, O-RAN Alliance, Alfter, Germany, 2025.