Proactive Resource Optimization for Heterogeneous 5G Network Slicing

Jesutofunmi Ajayi*, Torsten Braun*
*Institut für Informatik, Universität Bern, Switzerland
{firstname.lastname}@unibe.ch

***Abstract*—Network slicing enables 5G and beyond networks to concurrently support diverse services, including emerging IoT applications, over a shared, softwarized infrastructure. Traditional resource management approaches are often reactive, resulting in suboptimal utilization and potential Service Level Agreement (SLA) violations under dynamic traffic conditions. We present Proactive Resource OPtimization for Heterogeneous nETwork Slicing (PROPHET), a framework for *Proactive Resource Optimization* in heterogeneous slice-enabled networks. PROPHET integrates an attention-based traffic forecasting model with a deep reinforcement learning (DRL) resource allocator based on Proximal Policy Optimization (PPO) to anticipate future slice demands and proactively adjust configurations. Evaluated using real LTE traffic traces representing enhanced Mobile Broadband (eMBB), ultra-Reliable and Low-Latency Communication (uRLLC)-like, and best-effort services, PROPHET improves SLA compliance and resource efficiency compared to reactive baselines. The proposed framework is equally applicable to IoT-driven slices, supporting massive connectivity and ultra-reliable low-latency requirements envisioned for next-generation smart environments.**

***Index Terms*: Network Slicing, LSTM, Proactive Optimization, Resource Allocation, 5G**

I. Introduction

5G networks support diverse applications and services ranging from enhanced mobile broadband to massive machine-type communications for emerging Internet-of-Things (IoT) verticals [1]. However, managing heterogeneous network traffic with varying Quality of Service (QoS) requirements and dynamic demands challenges traditional static or reactive resource allocation strategies [2], [3], leading to suboptimal utilization and potential SLA violations.

Network Slicing (NS) has emerged as a key 5G enabler, creating isolated virtual networks tailored to specific application requirements on shared infrastructure. However, this requires dynamic, intelligent resource management capable of adapting to fluctuating Network Slice (NSL) demands. Current coarse-grained approaches lack the adaptability and granularity needed for diverse performance requirements in evolving network environments[4].

The shift toward virtual and open Radio Access Network (RAN) architectures introduces disaggregated mobile networks which offer greater flexibility and vendor diversity, but demand sophisticated resource orchestration across network components, such as the Central Unit (CU), Distributed Unit (DU) and Radio Unit (RU) [5], [6], [7]. These resource orchestration solutions must optimize resources to guarantee NSL requirements despite unpredictable load variations, preventing resource inefficiency, SLA violations, and QoS deterioration. To achieve such dynamic network slice management, proactive resource optimization techniques based on Machine Learning (ML) mechanisms are required to enable efficient management of NSL resources [8]. Specifically, we aim to address the following research questions:

- How can ML-based traffic forecasting be integrated with reinforcement learning to enable proactive resource allocation in heterogeneous 5G network slicing environments?
- To what extent does the forecasting accuracy and prediction horizon of a Deep Learning (DL) model impact the effectiveness of proactive resource allocation strategies in maintaining SLAs while improving resource efficiency?

To address these questions, this paper proposes PROPHET, a framework for proactive resource optimization in slice-enabled networks. PROPHET combines DL for traffic forecasting with Deep Reinforcement Learning (DRL) for resource allocation. Specifically, PROPHET integrates an attention-based DL model to predict traffic demands across network slice instances, then employs a DRL agent to proactively allocate resources based on these predictions. This approach anticipates future resource needs, scales the capacity of NSLs in advance, and optimizes resource utilization while meeting diverse QoS requirements through learned allocation policies. Our main contributions can be summarized as follows:

- Design of an attention-based DL model for predicting network traffic demands across heterogeneous services in slice-enabled 5G environments.
- Development of a DRL framework that learns optimal resource allocation policies for heterogeneous traffic flows while adapting to dynamic network conditions.
- Performance evaluation through simulations demonstrating improved resource utilization and reduced SLA violations, captured by the agent's reward, compared to baseline methods.

The remainder of this paper is organized as follows. In the next section we present the related works. In Section III, we present the system model and formulate the considered resource allocation problem. We describe the components of the proposed PROPHET framework in Section IV. In Section V, we evaluate our proposed approach in terms of prediction accuracy, SLA violations, and resource utilization. We conclude the paper in Section VI.

ISBN: 978-3-903176-79-9

II. Related Work

Abiko et al. [9] proposed a DRL solution to dynamically allocate Physical Resource Blocks (PRBs) to RAN slices, thereby enhancing profits and efficiency. Leconte et al. [10], propose a flexible and lightweight resource allocation framework, which uses cloud-native architectural concepts, and methods from continuous optimization, to provision and auto-scale slices in real-time. Their approach is based on flexible utility functions that are subject to the network bandwidth and processing power capacity of cloud instances.

A data-driven resource optimization approach is proposed by Mahmoud et al. [11] and tested in an Open Radio Access Network (O-RAN) environment. In their approach, they evaluated various algorithms based on their effectiveness in minimizing the number of PRBs used in the system by optimizing the Throughput-to-Bandwidth ratio.

Yeh et al. [12] propose an intelligent network application (or xApp) for slicing the RAN using AI and Deep Learning techniques. Through their evaluation, the authors show how different services can co-exist over a common network infrastructure, while meeting the SLAs of each service.

Kak et al. [13] develop an automatic Network Slicing framework to address the objectives of computing optimal network routes and allocating network resources, with minimal SLA violations. Their proposed framework does not utilize prior information concerning the resource requirements associated with a NSL, and is therefore robust to a wide variety of use cases and scenarios.

Mohanti et al. [14] propose a resource allocation framework that leverage predictions of the network quality. Their proposed solution demonstrates how look-ahead forecasting of channel conditions can improve the performance of the network over traditional resource allocation methods.

Sciancalepore et al. [15] propose a resource allocation framework for 5G network slicing, consisting of: traffic forecasting per slice using Holt-Winters forecasting, heuristic-based admission control, and adaptive forecast correction through a dynamic slice schedule.

Fatmeh et al. [16] proposed a joint intelligent framework for traffic forecasting, flow-split distribution, dynamic user association, and radio resource management within the disaggregated O-RAN architecture. They divide the optimization problem into long-term and short-term subproblems, where the long-term component uses an Long Short-Term Memory (LSTM) model for traffic forecasting and flow-split decisions, while the short-term component is efficiently solved using successive convex approximation technique.

Van Huynh et al. [17] developed an intelligent resource slicing framework that efficiently handles real-time network dynamics and uncertain user demands. They model the resource optimization problem using a semi-Markov decision process and apply Q-learning to derive the optimal policy without requiring prior knowledge of environment parameters.

Wei et al. [18] propose a predictor-optimizer framework that intelligently performs inter-slice reconfiguration with the aim of minimizing the energy consumption while provisioning NSLs. They use prediction intervals that are made of lower and upper bounds which constrain the future traffic demands with a pre-specified probability.

TABLE I: Related Works
TF = Traffic Forecasting, HS = Heterogeneous Slices
PO = Proactive Optimization, IF = Integrated Framework

Reference	HS	TP	PO	IF
Abiko et al. [9]	✓	✗	✓	✗
Alchaab et al. [19]	✓	✓	✓	✓
Fatmeh et al. [16]	✓	✓	✗	✗
Kak et al. [13]	✓	✗	✗	✗
Leconte et al. [10]	✓	✗	✗	✗
Mahmoud et al. [11]	✓	✗	✗	✗
Mohanti et al. [14]	✗	✓	✗	✗
Rajak et al. [20]	✓	✓	✗	✗
Scianlepore et al. [15]	✗	✓	✗	✗
Van Huynh et al. [17]	✓	✗	✓	✗
Wei et al. [18]	✗	✓	✗	✗
Yeh et al. [12]	✓	✓	✗	✗
Current work	✓	✓	✓	✓

Alchaab et al. [19] proposed an algorithm for resource allocation optimization in O-RAN to enhance QoS, by jointly solving a long term resource allocation problem and a short-term resource scheduling problem. Their solution, which is a Long Short-Term Memory Actor-Critic-based algorithm, is a modification of the LSTM and Deep Deterministic Policy Gradient (DDPG) algorithms. They show that their proposed approach outperforms state-of-the-art schemes and significantly reduces the utilization of network resources.

Rajak et al. [20] propose an attention-based LSTM technique for efficient slice classification. Their approach achieves high classification accuracy, enhancing both quality of service and quality of experience, while enabling service providers to meet strict SLAs. A summary of the related works can be found in Table I.

The key difference between previous works and our proposed approach is that we develop an integrated framework that seeks to optimize the performance of heterogeneous NSLs through predicting upcoming traffic and learning a proactive resource allocation policy. Unlike state-of-the-art methods that often rely on reactive adjustments or separate heuristic-based controls, PROPHET tightly couples attention-based forecasting with a DRL decision-making process. Specifically, we utilize a hybrid attention-LSTM model to capture complex temporal dependencies, directly informing a PPO agent to scale the capacity of NSLs in advance. This integration enables the system to address the specific Quality of Service (QoS) requirements of heterogeneous slices, such as eMBB and URLLC-like services, which typically challenge coarse-grained allocation strategies. Furthermore, by embedding forecasted demands directly into the state space, our approach shifts the paradigm from error correction to violation prevention, significantly reducing the probability of SLA violation under dynamic traffic conditions.

III. System Model and Problem Formulation

In this section, we present the system model as well as the formulation of the resource allocation problem.

A. System Model

We consider a network slicing scenario comprised of a single Base Station (BS), where a set of NSLs $\mathcal{H} = \{h_1, h_2, \ldots, h_n\}$ must compete for a limited amount of network resources (*i.e.,* Physical Resource Blocks (PRBs)). Each NSL $h \in \mathcal{H}$ is characterized by a traffic demand d_h, a QoS requirement (*e.g.,* maximum acceptable jitter $J_h^{\max}$), and a priority weight w_h that reflects its relative importance in the system. To meet the QoS requirements for each NSL, resources a_h within a bounded range must be allocated and are defined by a minimum allocation $a_h^{\min}$ and a maximum allocation $a_h^{\max}$, which yields the following inequality:

$$a_h^{\min} \le a_h \le a_h^{\max} \quad \forall h \in \mathcal{H} \tag{1}$$

The total available amount of PRBs are limited to K, and resource allocation decisions must ensure that the cumulative allocation across all NSLs does not exceed this capacity. This constraint is represented as:

$$\sum_{h=1}^{|\mathcal{H}|} a_h \le K \tag{2}$$

Additionally, the jitter experienced by a NSL is influenced by its allocated resources. For each NSL h, we define a binary variable $v_h \in \{0, 1\}$ to indicate whether an SLA violation occurs, where $v_h = 1$ represents a violation and $v_h = 0$ indicates SLA compliance. A non-negative variable $\delta_h \ge 0$ effectively quantifies the absolute difference between the total traffic demand of NSL h and the total amount of resources allocated to it. This variable reflects the degree of fairness, or deviation, in how network resources are distributed compared to actual demand, providing a measure of resource allocation inequality across network slices.

B. Problem Formulation

The objective of the optimization problem is to maximize the network's objective function by balancing NSL satisfaction and resource efficiency. Specifically, the goal is to maximize the weighted sum of SLA satisfied slices, represented by $\sum_{h \in \mathcal{H}} w_h(1 - v_h)$, while minimizing the overall deviation between traffic demand and resource allocation, captured by $\sum_{h \in \mathcal{H}} \delta_h$, where w_h represents the priority weight of NSL h. The resulting objective function ensures that high-priority NSLs are favored, SLA violations are penalized, and fairness in resource allocation is encouraged.

SLA violations are monitored via a jitter detection model, where it is assumed that the observed network jitter is inversely proportional to the PRBs allocated to each specific NSL. Specifically, this model aims to identify resource allocation inefficiency by evaluating the ratio of traffic demand to the allocated capacity. This relationship is modeled as:

$$\frac{d_h}{a_h + \epsilon} \le J_s^{\max}$$

where ϵ is a small constant to avoid division by zero. If this condition is not satisfied for a given NSL, it is considered an SLA violation. This deterministic model formulation allows us to clearly identify when the allocated resources are insufficient to meet the jitter requirement. Specifically, the jitter model adopted is simplified and deterministic in order to capture the expected inverse relationship between allocated PRBs and the likelihood of SLA violations in the considered setting. In practical RAN deployments, however, jitter typically arises from a combination of stochastic effects, including packet queueing dynamics, scheduling granularity, retransmissions, and time-varying channel conditions, which affects the network's performance and effectiveness [21]. Rather than explicitly modeling these low-level mechanisms, our model formulation uses a proxy based on the ratio between traffic demand and allocated capacity to indicate congestion-induced delay variation.

In our considered scenario, resource efficiency is evaluated by comparing the relative proportions of traffic demand and allocated resources. For each NSL h, the demand proportion p_h^d is defined as the fraction of total traffic demand attributed to it, while the allocation proportion p_h^a represents the fraction of total resources it receives. To quantify the mismatch between these proportions, we introduce the variable δ_h, which captures the absolute difference between demand and allocation $\delta_h = |p_h^d - p_h^a|$. This formulation ensures that even small mismatches between demand and allocation are accounted for, thereby promoting fairness and efficiency in resource distribution between the active NSLs. The objective of the considered problem is therefore given by:

$$\max \sum_{h \in \mathcal{H}} w_h(1 - v_h) - \sum_{h \in \mathcal{H}} \delta_h \tag{3}$$

$$\text{s.t. } (1)(2) \tag{4}$$

This problem can be formulated as a Markov Decision Process (MDP), where an agent dynamically selects actions based on the current network state with the goal of optimizing a given reward function that considers both SLA satisfaction and resource efficiency.

C. MDP Reformulation

To overcome the challenge of finding a solution to the formulated problem within a reasonable time, the problem is reformulated as a sequential decision problem within the framework of an MDP, where the goal is to learn a resource allocation policy that meets the objectives of the stated problem, while respecting the constraints. An MDP can generally be specified by a five-tuple which includes state space, action space, transition probability from the current state to the next, reward, and discount factor, i.e.,$\langle S, A, P, R, \gamma \rangle$, respectively.

- *State Space*: The state space represents the environment, which is defined as a set of states $s(t) \in S$, and represents the information available to the agent at the beginning of each time step t. We define this by: $s_t = \{d_h(t), a_h(t-1), v_h(t-1), c_h(t), f_h(t), K(t)\}_{h \in \mathcal{H}}$, where $d_h(t)$ represents the current traffic demand for NSL

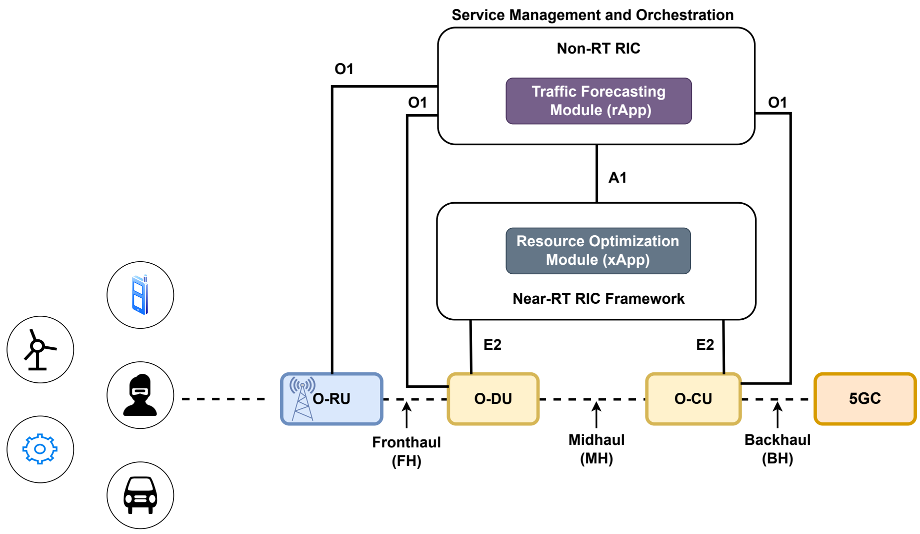

Fig. 1: PROPHET Framework in an O-RAN architecture

h, $a_h(t)$ represents the previous resources allocated to the NSL h, $v_h(t)$ represents the current SLA status of the slice, which includes the average jitter for the set of User Equipments (UEs) $\mathcal{U}$ in the NSL, the average Channel Quality Indicator (CQI) for the set of UEs in the slice is represented by $c_h(t)$, $f_h(t)$ represents the forecasted aggregated traffic demand for the NSL over a specified future horizon (*i.e.,* the demand at $t + w$, where w is the considered horizon) and, finally, the capacity of the resources is represented by $K(t)$.

- *Action Space*: In the considered environment, the action space defines the agent's behavior and is based on a policy π, which determines the agent's actions on the environment (*i.e.,* the resource allocation decision taken by the agent in a given state s_t). The DRL agent's action is given by the following vector $\mathbf{a}(t) = \{a_h(t)\}_{h \in \mathcal{H}}$, where $a_h(t)$ represents the proportion of total resources to be allocated to NSL h and $a_t \in A$. The environment processes this action a_t to derive the amount of resources (*i.e.,* PRBs) to effectively allocate by scaling the allocation proportions to the total available resources K (Eq. (2)) and applying per-slice minimum and maximum resource constraints (Eq. (1)).
- *Reward Function*: We define the reward as $r_t = R(s_t, a_t)$, which is received after taking action a_t in state s_t. Specifically, we compute the reward as: $r_t = \sum_{h \in \mathcal{H}} w_h(1 - v_h) - \sum_{h \in \mathcal{H}} \delta_h$ where w_h is the priority weight of the NSL h, $v_h = \mathbb{1}(\frac{d_h}{a_h + \epsilon} \geq J_s^{\max})$ is an indicator function that evaluates to 1 if the SLA is violated and 0 otherwise, and $\delta_h \geq |p_h^d - p_h^a|$ captures the resource efficiency of the action, with $p_h^d(t) = \frac{d_h(t)}{\sum_{h' \in \mathcal{H}} d_{h'}(t)}$ and $p_h^a(t) = \frac{a_h(t)}{\sum_{h' \in \mathcal{H}} a_{h'}(t)}$.

The goal of an agent is to find a policy $\pi : S \to A$ that maximizes the expected reward:

$$\pi^* = \arg\max_{\pi} \mathbb{E}_{\pi} \left[\sum_{t=0}^{\infty} \gamma^t r_t \right], \tag{5}$$

$\gamma \in [0, 1]$ controls the relative importance of immediate versus future rewards.

IV. PROPHET Framework

NSL traffic demands vary continuously due to the presence of dynamic users, applications, and services, requiring intelligent strategies that can accurately predict the spatial-temporal characteristics of traffic patterns for optimal resource allocation to NSLs [22], [23]. As a result, accurate traffic forecasting is crucial for efficient resource allocation, as it enables Infrastructure Providers (InPs) to optimize NSL resources and meet diverse SLAs, which is a critical goal of our proposed solution.

To maintain the QoS across the shared mobile network infrastructure, network operators must monitor the performance of NSLs throughout the network and dynamically reconfigure their capacity to prevent service degradation that would potentially violate SLAs of the active slices.

A. Attention-based LSTM Traffic Forecasting

To predict the traffic in the NSLs, we develop a traffic forecasting module that leverages a hybrid architecture combining Bidirectional LSTM layers with attention mechanisms to better capture the temporal dynamics of network traffic patterns [22]. The module plays a crucial role in enabling proactive decision-making for the DRL agent. As this module is responsible for capturing the temporal dynamics and traffic patterns of heterogeneous NSLs, it can be placed as an *rApp* at the Non-Real Time RAN Intelligent Controller (Non-RT RIC) in an O-RAN architecture, as shown in Fig. 1.

Based on the MDP formulation, at each time step t, after the agent has taken an action and the current environment state (including current traffic demand $\mathbf{d}(t)$) has been observed, the traffic forecasting component is invoked. The forecaster employs a sequence-to-sequence approach with a configurable sequence length of 10 time steps, utilizing MinMaxScaler normalization to ensure stable training. It processes historical traffic data up to time t to generate predictions $f_h(t)$ for each NSL h over the next w time steps (where w represents the forecast horizon). The architecture consists of stacked bidirectional LSTM layers (25 units each) with dropout regularization (0.3) for capturing long-range dependencies and avoiding overfitting, followed by dense layers for final prediction [24]. This design choice enables the model to learn both forward and backward temporal patterns in the traffic data, which is particularly important for capturing the often cyclical nature of network usage patterns.

B. DRL-based Resource Optimization

Based on the MDP formulation, we leverage a DRL algorithm to learn a data-driven resource allocation policy. More specifically, in the considered MDP, the agent explores the environment by taking actions in several states, without having apriori knowledge about which actions are more beneficial or optimal, with the goal of eventually learning the best policy through experience [25]. As the optimization module contains the DRL agent that determines the resource allocation of the heterogeneous network slices, it can be deployed as an intelligent *xApp* at the Near-Real Time RAN Intelligent Controller (Near-RT RIC) (Fig. 1).

To learn a proactive resource allocation policy, we employ the PPO algorithm [26], a state-of-the-art DRL method known for its stability and sample efficiency. PPO iteratively refines a policy $\pi_\theta(a_t|s_t)$ and a value function $V_\phi(s_t)$ parameterized by neural networks with parameters θ and ϕ, respectively. The policy network maps the current comprehensive state s_t which includes current network information (*i.e.,* actual traffic, observed jitter and CQI), historical data (past resource allocations, traffic, and QoS metrics), and critically, forecasted traffic demands f_h for a future horizon w to a distribution over possible resource allocation actions a_t.

The learning process involves collecting batches of experience by allowing the agent to interact with the simulated network environment using its current policy. For each state-action pair, the advantage A_t is estimated using Generalized Advantage Estimation (GAE), which quantifies how much better the chosen action was compared to the policy's average performance from that state. Then, PPO updates it's policy parameters θ by maximizing a clipped surrogate objective function:

$$L^{\text{CLIP}}(\theta) = \mathbb{E}\Big[\min\Big(r_t A^t,\ \text{clip}(r_t, 1-\epsilon, 1+\epsilon)A^t\Big)\Big], \qquad \text{where } r_t = \frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)}. \tag{6}$$

This objective encourages policy improvement while constraining the update step size via the clipping mechanism (controlled by ϵ) preventing destructive large updates. Concurrently, the value function parameters ϕ are updated by minimizing the mean squared error between $V_\phi(s_t)$ and the empirical return.

The proactive behavior of the learned policy primarily arises from including forecasted traffic into the state s_t. By observing predicted future demands, the agent learns to anticipate resource needs and adjust allocations preemptively, rather than simply reacting to current networking conditions. The PPO algorithm, through its value function and advantage estimation effectively attributes responsibility for long-term outcomes, such as SLA violations or efficient PRB resource utilization resulting from responding to forecasts, enabling it to learn sophisticated, proactive resource allocation policies that approximate a solution to the constrained optimization problem in Eq. (3).

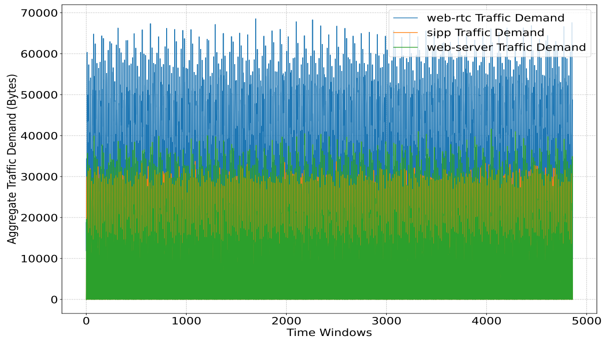

Fig. 2: Per slice aggregate traffic demands

Dataset Feature	Description
Current Demand	Traffic for considered applications: WebRTC, SIPp, Web-Server
CQI	Channel Quality Indicator reflecting the average quality of the radio link
Jitter	UE experienced Jitter (s)
Output	**Description**
Predicted Demand	Future traffic demand per NSL

TABLE II: Dataset Features and Output

V. Evaluation

We empirically evaluate the performance of our proposed PROPHET framework. We first describe the simulation setup, including the dataset and performance metrics. Subsequently, we present and analyze the results, focusing on the efficacy of the attention-based traffic forecasts, the proactive resource allocation capabilities of the PPO agent in terms of SLA compliance, and overall resource utilization efficiency. We also compare PROPHET against relevant baseline approaches.

A. Dataset Preparation and Processing

Our experiments leverage a publicly available dataset [27] comprising real-world, high-frequency (1 Hz) network traffic and QoS statistics (Jitter, CQI per UE) from UEs operating in a commercial LTE network. This dataset captures traffic from distinct applications (WebRTC, SIP protocol (SIPp), Web-Server), which serve as *proxies* for three heterogeneous network slice types in our simulation: a real-time communication slice (e.g., eMBB/URLLC hybrid), a signaling/control slice (e.g., URLLC/mMTC), and a best-effort data slice (e.g., eMBB). To align with practical DRL agent decision-making intervals, we apply temporal aggregation to the raw 1 Hz data. The dataset is partitioned into contiguous, non-overlapping time windows of W_s, where $W_s = 10$. Within each window, traffic volumes for UEs belonging to the same service type are summed to compute the aggregate demand for each NSL. Key QoS metrics like jitter and CQI are averaged per UE within each window to yield representative performance indicators. This processing yields the temporally aggregated dataset used as input for our simulation environment, as illustrated in Fig. 2.

B. Simulation Settings

Hyperparameter	Configuration
Multi-Head Attention Layers	1
Number of Heads	4
Dimensionality of Attention Vectors	64
LSTM Layers	1
LSTM Units	25
Units per Hidden Layers	16
Dense Layers	1
Optimizer	Adam
Learning Rate	10^{-3}
Loss Function	MSE
Activation Function	ReLU

TABLE III: Attention-LSTM Parameters

Hyperparameter	**Value**
Learning Rate	3×10^{-4}
Steps per Update	1024
Batch Size	64
Number of Epochs	10
Discount Factor (γ)	0.99
GAE Lambda (λ)	0.95
Clip Range (ϵ)	0.2
Entropy Coefficient	0.01
Value Function Coefficient	0.5
Max Gradient Norm	0.5

TABLE IV: PPO Hyperparameters

The attention-based forecasting model (Section IV-A) is trained on the initial 70% of the aggregated dataset to predict per-slice traffic demands w steps ahead, where $w = 5$ windows. Specifically, we use the Mean Squared Error (MSE) loss with the Adam optimizer and incorporate early stopping to speed up the training process and avoid overfitting. The specific details about the hybrid architecture and the hyperparameters used are given in Table III and are generated using grid search for hyperparameter tuning. The remaining 30% is used for evaluating the DRL agent. The PPO agent is configured with the parameters given in Table IV. The three NSLs (*i.e.,* WebRTC, SIPp, Web-Server) are configured with specific SLA targets for maximum jitter and minimum effective CQI, and resource allocation bounds (*i.e.,* a_h^{min} and a_h^{max}), as outlined in Section III-A. We consider a 20 MHz configuration for the BS, which translates to 100 PRBs available to be allocated between the NSLs, based on which, we set the values of a_h^{min} and a_h^{max} to 10 PRBs and 30 PRBs, respectively. The resource allocation constraints reflect the heterogeneous requirements of different service types in the network.

The simulation network environment evolves according to a deterministic and reproducible state transition process at each time step. At time t, the DRL agent outputs an action vector $\mathbf{a}(t) = \{a_h(t)\}_{h \in H}$ representing per-slice resource allocation proportions. These proportions are scaled to the total number of available PRBs, clipped according to slice-specific minimum and maximum bounds, and normalized to satisfy the total resource constraint. Traffic demands $d_h(t)$ are then realized from the temporally aggregated dataset window corresponding to time t. For each NSL h, a jitter proxy is computed as a function of the demand-to-allocation ratio, and the SLA violation indicator v_h is derived by comparing this value against the slice-specific jitter threshold. Channel quality indicators for the next state are obtained by averaging per-UE CQI values from the dataset for the same time window. The next state $s(t+1)$ is subsequently formed by combining updated traffic demands, CQI values, SLA indicators, previous allocations, and forecasted future demands. Finally, the reward is computed according to Eq. (3), completing the transition to the next decision step.

Despite the simplicity of the adopted jitter proxy, stochasticity is implicitly present in the environment through multiple dimensions. First, traffic demands are derived from publicly available LTE/5G traffic traces described in Section V-A, which exhibit temporal variability and mild burstiness, as illustrated in Fig. 2. Second, channel quality fluctuations are reflected through time-varying CQI values included in the agent's state representation. Third, the delayed impact of resource allocation decisions is captured through the Markov decision process formulation and the discounting of future rewards, allowing the agent to account for longer-term consequences of its actions.

1) Baselines: To evaluate the advantage of our proactive solution, we compare the performance of our approach against a PPO agent that is identical in architecture and hyperparameters, but without access to traffic forecasts in it's state space. Hence, the agent makes decisions based only on current and historical observations, and is considered as a reactive baseline solution.

2) Performance Metrics: We evaluated the performance of our approach by considering the following metrics:

- **Traffic Forecasting Accuracy** is evaluated using Normalized Root Mean Square Error (NRMSE) for each NSL/service type.
- **Average Cumulative Reward** is the average sum of discounted rewards obtained by the DRL agent per episode during evaluation.

C. Results and Analysis

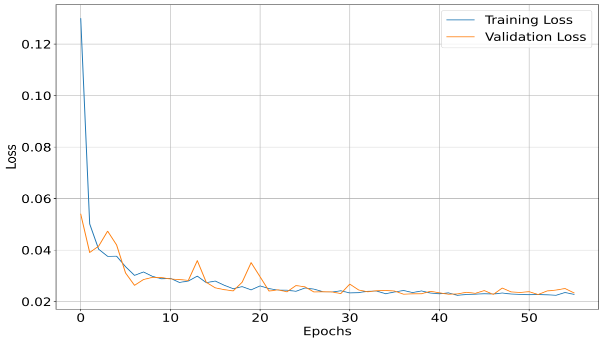

Fig. 3: Training and Validation Loss

Service Type	NRMSE
WebRTC	21.00%
SIPp	13.15%
Web-Server	14.50%

TABLE V: Forecasting Performance Metrics

1) Traffic Forecasting: The consistent decline and convergence of the training and validation loss are shown in Fig. 3. The figure demonstrates that the attention-based DL model is well fitted to the overall dataset constructed for the purpose of our simulations and converges after 50 epochs. Specifically, the hybrid architecture, which combines Bidirectional LSTM layers with multi-head attention mechanisms, enables the model to learn both forward and backward temporal patterns in the traffic data of the dataset. This capability is particularly important for accurately capturing the often dynamic nature of network traffic patterns [28]. The model was trained based on the MSE loss and the Adam optimizer. The forecasting performance of traffic forecasting model, in terms of accuracy, is evaluated on the remaining 30% of the aggregated dataset, and is presented in Table V and shows stable performance of the traffic forecaster across the different service types which serve as proxies for heterogeneous 5G NSLs. Specifically, the attention-based DL model in the forecasting model achieved relatively low NRMSE values of 21%, 13.15%, and 14.5% for WebRTC, SIPp, and Web-Server traffic, respectively, indicating that the hybrid architecture effectively captures the temporal dynamics of the heterogeneous network traffic patterns within each NSL. As a result of the stable performance of the DL model, we are able to leverage the model in our simulations to predict future traffic and states to aid the DRL agent in making proactive resource allocation decisions that minimize SLA violations and optimize for resource efficiency.

2) Average Cumulative Reward: We evaluated the performance of the proposed PROPHET framework against relevant baselines by analyzing the Average Cumulative Reward obtained by the DRL agent per episode during the evaluation. The reward function is designed to maximize the weighted sum of satisfied slices while minimizing the deviation between traffic demand and resource allocation, thereby heavily penalizing SLA violations and improving resource efficiency. As shown in Fig. 4, when configured with an optimal forecasting horizon (w = 5), PROPHET significantly outperforms the reactive baselines and achieved a better average cumulative average reward (approx. average: -516) compared to w = 20 (approx. average: -1040) and significantly outperforms the standard PPO baseline (approx. average: -752) and the Soft-Actor Critic (SAC) learning baseline (approx. average: -936). The improvement demonstrated by PROPHET (w = 5) over the standard PPO and SAC baselines is attributed to it's proactive optimization strategy. Unlike the reactive baselines, which make decisions based only on current and historical observations, PROPHET integrates a DL model to predict future traffic demands of the heterogeneous NSLs in the environment. The DRL agent's state space (s_t) critically includes forecasted aggregated traffic demands (f_h) for a future horizon (w). By observing these predicted demands, the DRL agent learns policies that allow it to anticipate resource needs and adjust the slice allocations in a preemptive manner. The ability to preemptively scale the capacity of NSLs results in reduced SLA violations and improved cumulative rewards compared to baselines that can only react to service degradation after it has already occurred. However, we also observe that the performance of the framework is significantly reduced when the forecasting horizon is longer (w = 20), which is attributed to the fact that longer forecasting horizons could potentially introduce excessive forecast uncertainty or include irrelevant and incorrect future information, leading to larger penalties for the DRL agent [29].

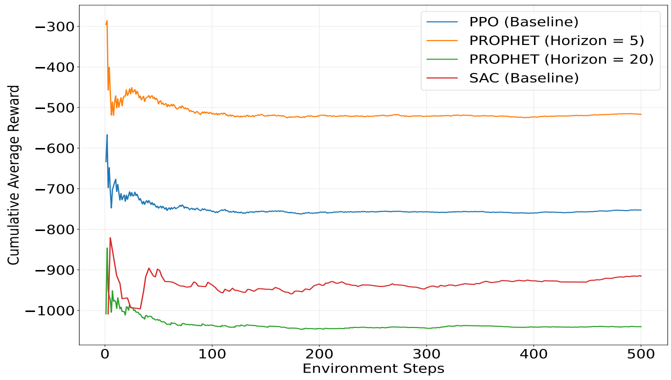

Fig. 4: Cumulative Average Reward: PROPHET vs Baselines

VI. Conclusion

We introduced the PROPHET framework for proactive resource optimization in 5G environments with network slices that support heterogeneous services. While evaluated on eMBB or uRLLC-like traffic services, PROPHET's proactive optimization framework can be equally applied to other slice types, including IoT-driven network slices. By integrating a attention-based traffic forecasting model with a PPO-based DRL agent, we demonstrated a systematic approach to anticipatory resource allocation that addresses the fundamental challenge of dynamic traffic management in slice-enabled mobile networks. Our framework leverages multi-dimensional state representations, including current network conditions and traffic forecasts, to enable the DRL agent to learn allocation policies that balance SLA compliance with resource efficiency.

Through comprehensive evaluation using real-world network traffic data, we observed that the forecasting model achieved stable NRMSE values of 21%, 13.15%, and 14.50% for WebRTC, SIPp, and Web-Server traffic, respectively. While these forecasting accuracies indicate room for improvement, the integrated PROPHET framework still demonstrated a better average cumulative reward which encapsulates it's resource efficiency and SLA violation policy, compared to reactive baseline methods. The framework successfully maintains service differentiation, enforcing slice-specific resource bounds while adapting to the heterogeneous QoS requirements of different service types.

REFERENCES

[1] K. Dang, H. Khalifé, M. Sintorn, D. Lindbo, and S. Secci, "Deep Reinforcement Learning for Joint Energy Saving and Traffic Handling in xG RAN," in *ICC 2024 - IEEE International Conference on Communications*, 2024, pp. 4743–4748.

[2] S. Wu, N. Chen, A. Xiao, P. Zhang, C. Jiang, and W. Zhang, "AI-Empowered Virtual Network Embedding: A Comprehensive Survey," *IEEE Communications Surveys & Tutorials*, 2024.

[3] S. Vassilaras, L. Gkatzikis, N. Liakopoulos, I. N. Stiakogiannakis, M. Qi, L. Shi, L. Liu, M. Debbah, and G. S. Paschos, "The Algorithmic Aspects of Network Slicing," *IEEE Communications Magazine*, vol. 55, no. 8, pp. 112–119, 2017.

[4] W. Rafique, J. Rani Barai, A. O. Fapojuwo, and D. Krishnamurthy, "A Survey on Beyond 5G Network Slicing for Smart Cities Applications," *IEEE Communications Surveys & Tutorials*, vol. 27, no. 1, pp. 595–628, 2025.

[5] E. Schiller, J. Ajayi, S. Weber, T. Braun, and B. Stiller, "Toward a Live BBU Container Migration in Wireless Networks," *IEEE Open Journal of the Communications Society*, vol. 3, pp. 301–321, 2022.

[6] C. P. Galdino, R. S. Couto, D. S. De Medeiros, I. M. Moraes, and D. M. Mattos, "Reinforcement Learning for Radio Resource Management in O-RAN," in *2024 IEEE 13th International Conference on Cloud Networking (CloudNet)*. IEEE, 2024, pp. 1–9.

[7] S. D. A. Shah, M. Hafeez, A. Salama, and S. A. R. Zaidi, "Proactive AI-and-RAN Workload Orchestration in O-RAN Architectures for 6G Networks," *IEEE Open Journal of the Communications Society*, vol. 6, pp. 7939–7954, 2025.

[8] M. M. H. Qazzaz, L. Kulacz, A. Kliks, S. A. Zaidi, M. Dryjanski, and D. McLernon, "Machine Learning-based xApp for Dynamic Resource Allocation in O-RAN Networks," in *2024 IEEE International Conference on Machine Learning for Communication and Networking (ICMLCN)*, 2024, pp. 492–497.

[9] Y. Abiko, T. Saito, D. Ikeda, K. Ohta, T. Mizuno, and H. Mineno, "Flexible Resource Block Allocation to Multiple Slices for Radio Access Network Slicing Using Deep Reinforcement Learning," *Ieee Access*, vol. 8, pp. 68 183–68 198, 2020.

[10] M. Leconte, G. Paschos, P. Mertikopoulos, and U. Kozat, "A Resource Allocation Framework for Network Slicing," in *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications*, 04 2018, pp. 2177–2185.

[11] H. Mahmoud, M. N. I. Farooqui, D. Mi, L. Guo, C. Lu, Y. Gan, Z. Gao, Z. Wang, and Y. Zhang, "Data-driven Approach for Optimising Resource Allocation of O-RAN Networks," in *2024 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2024, pp. 1–6.

[12] S.-p. Yeh, S. Bhattacharya, R. Sharma, and H. Moustafa, "Deep Learning for Intelligent and Automated Network Slicing in 5G Open RAN (ORAN) Deployment," *IEEE Open Journal of the Communications Society*, 2023.

[13] A. Kak and I. F. Akyildiz, "Towards Automatic Network Slicing for the Internet of Space Things," *IEEE Transactions on Network and Service Management*, vol. 19, no. 1, pp. 392–412, 2021.

[14] S. Mohanti, A. Malhotra, M. Umar, B. Farooq, and J. Chandrashekar, "PROMPT : Prediction of Channel Metrics for Proactive Optimization in Cellular Networks," *2024 IEEE 25th International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pp. 370–376, 2024.

[15] V. Sciancalepore, K. Samdanis, X. Costa-Perez, D. Bega, M. Gramaglia, and A. Banchs, "Mobile Traffic Forecasting for Maximizing 5G Network Slicing Resource Utilization," in *IEEE INFOCOM 2017 - IEEE Conference on Computer Communications*, 2017, pp. 1–9.

[16] F. Kavehmadavani, V.-D. Nguyen, T. X. Vu, and S. Chatzinotas, "Intelligent Traffic Steering in Beyond 5G Open RAN Based on LSTM Traffic Prediction," *IEEE Transactions on Wireless Communications*, vol. 22, no. 11, pp. 7727–7742, 2023.

[17] N. Van Huynh, D. T. Hoang, D. N. Nguyen, and E. Dutkiewicz, "Real-Time Network Slicing with Uncertain Demand: A Deep Learning Approach," in *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, 2019, pp. 1–6.

[18] F. Wei, G. Feng, Y. Sun, Y. Wang, and S. Qin, "Proactive Network Slice Reconfiguration by Exploiting Prediction Interval and Robust Optimization," in *GLOBECOM 2020-2020 IEEE Global Communications Conference*. IEEE, 2020, pp. 1–6.

[19] A. Alchaab, A. Younis, and 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)*, 2025, pp. 51–60.

[20] A. Rajak and R. Tripathi, "An Attention-enhanced LSTM Model for Efficient Network Slicing in Beyond 5G Networks," *Ad Hoc Networks*, p. 104070, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S157087052500318X

[21] N. Phuc Tran, O. Delgado, and B. Jaumard, "Proactive Service Assurance in 5G and B5G Networks: A Closed-Loop Algorithm for End-to-End Network Slices," *IEEE Transactions on Network and Service Management*, vol. 23, pp. 668–680, 2026.

[22] O. Stenhammar, G. Fodor, and C. Fischione, "A Comparison of Neural Networks for Wireless Channel Prediction," *IEEE Wireless Communications*, vol. 31, no. 3, pp. 235–241, 2024.

[23] T. Ouyang, K. Zhao, G. Hong, X. Zhang, Z. Zhou, and X. Chen, "Dynamic Edge-Centric Resource Provisioning for Online and Offline Services Co-Location via Reactive and Predictive Approaches," *IEEE Transactions on Networking*, pp. 1–16, 2025.

[24] Z. Zhao, N. Emami, H. Santos, L. Pacheco, M. Karimzadeh, T. Braun, A. Braud, B. Radier, and P. Tamagnan, "Reinforced-LSTM Trajectory Prediction-Driven Dynamic Service Migration: A Case Study," *IEEE Transactions on Network Science and Engineering*, vol. 9, no. 4, pp. 2786–2802, 2022.

[25] M. Tsampazi, S. D'Oro, M. Polese, L. Bonati, G. Poitau, M. Healy, M. Alavirad, and T. Melodia, "PandORA: Automated Design and Comprehensive Evaluation of Deep Reinforcement Learning Agents for Open RAN," *IEEE Transactions on Mobile Computing*, vol. 24, no. 4, pp. 3223–3240, 2025.

[26] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal Policy Optimization Algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[27] T. Tsourdinis, I. Chatzistefanidis, N. Makris, and T. Korakis, "UE Network Traffic Time-Series (Applications, Throughput, Latency, CQI) in LTE/5G Networks," 2022. [Online]. Available: https://dx.doi.org/10.21227/4ars-fs38

[28] H. B. Tsegaye, P. M. Tshakwanda, Y. M. Worku, C. Sacchi, C. Christodoulou, and M. Devetsikiotis, "LSTM-Based Resource Prediction for Disaggregated RAN in 5G Non-Terrestrial Networks," in *2024 IEEE Virtual Conference on Communications (VCC)*, 2024, pp. 1–6.

[29] A. Balasingam, M. Kotaru, and P. Bahl, "Application-Level Service Assurance with 5G RAN Slicing," in *21st USENIX Symposium on Networked Systems Design and Implementation (NSDI 24)*. Santa Clara, CA: USENIX Association, Apr. 2024, pp. 841–857. [Online]. Available: https://www.usenix.org/conference/nsdi24/presentation/balasingam