Moving Target Defense based Secured Network Slicing System in the O-RAN Architecture

Mojdeh Karbalaee Motalleb[†], Chafika Benzaïd[*], Tarik Taleb[*], Vahid Shah-Mansouri[†]
Email: {mojdeh.karbalaee,vmansouri}@ut.ac.ir, {chafika.benzaid, tarik.taleb}@oulu.fi
[†]School of ECE, University of Tehran, Tehran, Iran
[*]University of Oulu, Oulu, Finland

***Abstract*—The open radio access network (O-RAN) architecture's native virtualization and embedded intelligence facilitate RAN slicing and enable comprehensive end-to-end services in post-5G networks. However, any vulnerabilities could harm security. Therefore, artificial intelligence (AI) and machine learning (ML) security threats can even threaten O-RAN benefits. This paper proposes a novel approach to estimating the optimal number of predefined VNFs for each slice while addressing secure AI/ML methods for dynamic service admission control and power minimization in the O-RAN architecture. We solve this problem on two-time scales using mathematical methods for determining the predefined number of VNFs on a large time scale and the proximal policy optimization (PPO), a Deep Reinforcement Learning algorithm, for solving dynamic service admission control and power minimization for different slices on a small-time scale. To secure the ML system for O-RAN, we implement a moving target defense (MTD) strategy to prevent poisoning attacks by adding uncertainty to the system. Our experimental results show that the proposed PPO-based service admission control approach achieves an admission rate above 80% and that the MTD strategy effectively strengthens the robustness of the PPO method against adversarial attacks.**

***Index Terms*—Open Radio Access Network (O-RAN), Adversarial Attacks, Moving Target Defense (MTD).**

I. Introduction

A sixth generation (6G) wireless network will offer enhanced network capacity of 10Gbps/m^3, lower end-to-end latency below 1 ms, and increased data rates up to 1 Tbps. The 6G capabilities will unlock new applications and services, including holographic communications, wireless brain-machine interaction, autonomous driving, etc. [1].

6G networks will use network slicing to meet the varying QoS requirements of envisioned applications/services by dynamically creating logically isolated, service-tailored virtual networks (i.e., slices) on shared physical infrastructures [2]. A network slice instance consists of chained network functions and the required resources (e.g., compute, bandwidth, storage), spanning multiple technology domains (e.g., radio access network (RAN), core network (CN), and transport network). Despite its maturity in CN, network slicing remains challenging in other domains. The native virtualization and embedded intelligence of the open RAN (O-RAN) architecture are vital features to promote RAN slicing, enabling the delivery of genuinely end-to-end services to become a reality [3]–[5]. Specifically, O-RAN architecture introduces RAN Intelligent Controller (RIC). This software-defined network controller leverages the capabilities of Artificial Intelligence (AI) and Machine Learning (ML) to enable intelligent and closed-loop RAN resource management and optimization. The RIC is divided into non-real-time (Non-RT) RIC and near-real-time (Near-RT) RIC, which incorporate rApps and xApps, custom micro-service-based applications, operating on Non-RT scale ($> 1s$) and Near-RT scale ($10 - 1000ms$), respectively.

In RAN slicing, efficient slice-aware resource management through slice admission control is crucial. Recently, the potential of ML and, more particularly, Deep Reinforcement Learning (DRL) techniques have been explored for enabling optimal slice admission control strategies in the O-RAN system. A federated DRL is presented in paper [6] to manage multiple independent xApps in O-RAN for network slicing. Two xApps, which are jointly communicated, are implemented for power and physical resource block management. In [3], the problem of obtaining the optimal number of virtual network functions (VNFs) and baseband resource management is considered for the RAN slicing in the O-RAN architecture, which is solved using an optimization technique. In [7], an optimization technique for the infrastructure resource reservation is used to prioritize admission control for multiple slices.

ML techniques, including DRL, face vulnerabilities to adversarial attacks that manipulate data during (re)training or serving [8]–[10]. For example, resource manipulation could mislead a DRL-based slice admission model, wrongly rejecting RAN slice requests. Ensuring ML security is crucial for O-RAN integration, building trust in their decisions Addressing O-RAN's security from an ML perspective involves multiple methods: Zero Trust (ZT), blockchain, and Moving Target Defense (MTD). ZT adopts a proactive 'never trust, always verify' stance to fortify O-RAN infrastructure against ML threats. Blockchain ensures data integrity and transparent model history, enhancing trust and accountability in ML processes.An effective defense strategy discussed in [8] is the Moving Target Defense (MTD) paradigm. MTD enhances security by continually changing the attack surface, making it harder for attackers to predict and exploit vulnerabilities. This approach has gained in safeguarding ML models, particularly in computer vision and malware domains(e.g. [11], [12]).

The MTD technique provides a dynamic and proactive security approach, distinguishing it from ZT and blockchain, which primarily concentrate on access control and data integrity. MTD's continuous alteration of the attack surface poses a formidable challenge for adversaries, enhancing resilience against evolving threats. Consequently, our paper centers on employing MTD to secure the O-RAN system.In [13], the authors consider a Trojaning attack defense framework based on an MTD on the Deep neural network (DNN), which randomly selected dimensions in multidimensional training models. According to the results, they guarantee DNN's availability and protect it from Trojan attacks.

Currently, previous works focus on traditional RAN slicing with classic methods such as optimization and game theory or artificial intelligent/Machine learning (AI/ML) techniques. Although some of them highlight O-RAN slicing, they do not mention the softwarization of the O-RAN architecture in RAN slicing technology. Moreover, the lifecycle of slicing is not assumed in previous works. In contrast, our paper differs from current papers by focusing on the planning and creation phase of RAN slicing to estimate the optimal number of predefined VNFs in the O-RAN distributed unit (O-DU) and the central unit (O-CU) for different slices based on the processing delay threshold of the system, which is done on a large time scale. Afterward, we consider the managing phase of the RAN slicing lifecycle by analyzing the dynamic service admission control and power minimization on a small time scale for different services with different QoS in the processing layer of the O-RAN technology using the DRL technique. Due to the dynamic nature of the problem, the sequential decision-making, and the large state space, DRL is the most appropriate approach. Moreover, to the best of our knowledge, none of the existing contributions has considered the security issues stemming from using DRL techniques in the RAN slicing. The adversarial defense strategy employs MTD to bolster the proactive resilience of our DRL model against attacks. Implementing MTD entails training various models with similar performance for different xApps. The xApps will be randomly selected after learning. The main contributions of this paper are as follows:

- We study the problem of estimating the optimal number of VNFs in each slice and solving the secure dynamic service admission control and power consumption in the O-RAN using the RAN slicing.
- The problem of estimating the optimal pre-defined VNFs are solved mathematically on a large time scale to obtain how many VNF chains can be deployed for each slice. In contrast, the problem of service admission control and minimizing total power is solved dynamically on a small time scale using the PPO algorithm, an actor-critical DRL technique.
- We introduce a novel defense strategy that relies on the MTD paradigm to make the proposed DRL approach resilient to adversarial attacks to prevent degrading system utilization. Rather than shuffling the network as done in prior MTD studies, we shuffle the AI models for frequent system changes. The developed MTD strategy consists in dynamically picking a model from a set of PPO models trained with different configurations, increasing the adversary's uncertainty.
- The numerical results demonstrate the improvement in the PPO method against the baseline method and the negative impact of adversarial attacks on a PPO-based service admission control system without appropriate defenses. They also show the effectiveness of the novel MTD-based defense strategy in enhancing the solution's robustness against those attacks.

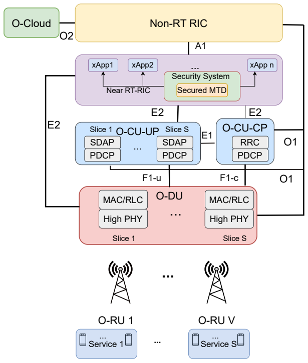

Fig. 1: The secured intelligent O-RAN architecture.

The rest of the paper is organized as follows. Section II introduces the system model. Section III formulates the target service admission control problem and presents the proposed DRL approach. Section IV describes the adversarial attack model and the devised MTD-based defense strategy. Section V and VI discuss the numerical results and conclusion.

II. System model

As illustrated in Fig. 1, we consider the dynamic slice admission control and power optimization in the O-RAN system. Assume there are S pre-defined RAN slices serving S services. Each instantiated RAN slice comprises several VNFs providing the services of the different virtualized O-RAN units, such as the O-DU and O-CU. Note that O-DU runs the high Physical (PHY), Medium Access Control (MAC), and Radio Link Control (RLC) layers. To deliver different services, MAC and RLC are deployed on isolated VNFs. The O-DU provides services to users through the radio unit (O-RU), which contains low PHY and radio frequency. The O-CU contains the O-CU control plane (O-CU-CP) and the O-CU user plane (O-CU-UP), which handle the control and data messages, respectively. The O-CU-CP includes packet data convergence protocol (PDCP) and radio resource control (RRC), and the O-CU-UP contains PDCP and service data adaptation protocol (SDAP) which are deployed on isolated VNFs for different services. As shown in Fig. 1, the virtualized O-RAN functions can be dedicated to each slice (e.g., O-DU, O-CU-UP) or shared between slices (e.g., O-CU-CP).

The O-RAN system uses Near-RT RIC and Non-RT RIC to control the O-DU and O-CU for resource management. The Near-RT RIC hosts third-party applications xApps to provide management and optimization services. Here, we consider the deployment of xApps providing DRL-based resource management services. The Non-RT RIC offers offline ML models to support Near-RT RIC functions.

We assume different services that use isolated pre-defined slices in this system model. The pre-defined slices contain reserved VNFs for two logical nodes of MAC/RLC functions in the O-DU and PDCP/SDAP functions in the O-CU-UP. We consider a simple service function chain in the O-DU and O-

CU. Suppose we have M_s^d and M_s^c VNFs in the O-DU and O-CU-UP processing layer for the service s.

A. Mean Delay

Consider the mean arrival rate of the service s is Poisson with rate $\bar{\alpha}_s^c$ in the O-CU-UP layer. The mean arrival data rate of the s^{th} service in the O-DU is approximately equal to the mean arrival data rate of the s^{th} service in the O-CU-UP ($\bar{\alpha} = \bar{\alpha}_s^d \approx \bar{\alpha}_s^c$). This is because the amount of data transmitted through the route (despite frame changes) is constant.

Incoming traffic to VNFs is divided equally by load balancers at each layer for each service. Assume that each VNF's baseband processing is represented by an M/M/1 queue. In each slice, one VNF processes each packet. Accordingly, the mean delay for slice $s$large time scale in the O-DU and O-CU can be calculated as M/M/1 queue [3] as follows as $\bar{T}_s^{DU} = \frac{1}{\bar{\mu}_s^d - \bar{\alpha}_s / M_s^d}$, and $\bar{T}_s^{CU} = \frac{1}{\bar{\mu}_s^c - \bar{\alpha}_s / M_s^c}$. In addition, $\frac{1}{\bar{\mu}_s^d}$ and $\frac{1}{\bar{\mu}_s^c}$ are the mean service time of the system in the O-DU and the O-CU-UP layer in the lrge time scale, respectively. For the simplicity, we assume that the O-CU and the O-DU have the same processing system. Hence $\frac{1}{\bar{\mu}_s^c} \approx \frac{1}{\bar{\mu}_s^d}$. Therefore, we can consider that the $M_s = M_s^d = M_s^c$, for the simplicity. As a result, $\bar{T}_s = \bar{T}_s^{DU} = \bar{T}_s^{CU}$. Consequently, the mean total delay of the system in the slice s is $\bar{T}_s^{tot} = 2 \times \bar{T}_s$.

B. Physical Data Center Resources

The VNF instances are also hosted on VMs that use data center resources. Each VNF in each layer requires specific physical rescurces including CPU, storage, and memory based on the service requirements. Consider a set of a tuple that expresses the instant required resources for VNF m in the service function chain of the $\mathfrak{z} \in \{c, d\}$ (VNFs of O-DU or O-CU) in slice s as $\bar{\psi}_s^{m^{\mathfrak{z}}} = \{\psi_{\mathsf{C},s}^{m^{\mathfrak{z}}}, \psi_{\mathsf{S},s}^{m^{\mathfrak{z}}}, \psi_{\mathsf{M},s}^{m^{\mathfrak{z}}}\}$. where, $\psi_{\mathsf{C},s}^m$, $\psi_{\mathsf{S},s}^m$, and $\psi_{\mathsf{M},s}^m$, provide the amount of CPU, storage, and memory that are required for the VNFs of the O-DU or O-CU ($\mathfrak{z} \in \{c, d\}$) . Moreover, $\bar{\psi}_s^m \in \mathbb{C}^3$. Accordingly, we indicate the total amount of CPU, storage, and memory, respectively ($\mathfrak{h} \in \{C, S, M\}$), for the O-DU and O-CU layers ($\mathfrak{z} \in \{c, d\}$) as $\bar{\psi}_{\mathfrak{h},s}^{\mathfrak{z},tot} = \sum_{m=1}^{M_s^{\mathfrak{z}}} \psi_{\mathfrak{h},s}^{m^{\mathfrak{z}}},\ \ \mathfrak{z} \in \{c, d\}, \mathfrak{h} \in \{C, S, M\}$.

Suppose we have N data centers for the VNFs of the O-DU and the O-CU. Each data center n, has a set of a tuple that expresses the amount of CPU, storage, and memory resources as $\chi_s^n = \{\chi_{\mathsf{C},s}^n, \chi_{\mathsf{S},s}^n, \chi_{\mathsf{M},s}^n\}$,. Assume, $x_{m_s^{\mathfrak{z}},n} \in \{0, 1\}$, is a binary variable describing whether the VNF $m_s^{\mathfrak{z}}$ in layer $\mathfrak{z} \in \{c, d\}$ in slice s is utilizing the data center n or not [14].

In the following, we will introduce an AI/ML method to optimize this system model. In addition, in this study, we consider a potential adversarial attack on our AI/ML approach which is a black-box attack (i.e., no knowledge). Attackers lack knowledge of our model and employ a weak adversary method to manipulate state and reward during agent interactions. Therefore, we require a secured technique to defend our system from these threats and vulnerabilities.

III. DRL-based Energy-efficient Service Admission Control

In this section, firstly the problem formulation is obtained. The proposed method is examined on two different time scales. An estimation of a pre-defined number of VNFs is achieved on a large time scale. Next, the dynamic admission control and power minimization are solved on a small time frame.

A. Problem Statement

Assume the priority of the service s is indicated with p_s. Moreover, for each data center n that is hosting the VNF $m_s^{\mathfrak{z}}$ in layer $\mathfrak{z} \in \{c, d\}$ in slice s, the power consumption of the baseband processing can be represented as $\phi_{m_s^{\mathfrak{z}},n}$. Therefore, the total power consumption of all running data centers that are hosting the VNFs can be expressed as $\phi_{tot} = \sum_{n=1}^{N} \sum_{s=1}^{S} \sum_{m_s^{\mathfrak{z}}=1}^{M_s^{\mathfrak{z}}} x_{m_s^{\mathfrak{z}},n} \phi_{m_s^{\mathfrak{z}},n},\ \ \mathfrak{z} \in \{c, d\}$. As a result, the cost function for the placement of VNFs on the data centers is formulated as follows:

$$\varphi_{tot} = \phi_{tot} - \kappa \sum_{n=1}^{N} \sum_{m_s=1}^{M_s} p_s x_{m_s,n} \tag{1}$$

Where κ is a design factor between the first term of (1), representing the whole power of the resources, and the second term, is the total number of slices admitted with resources. We aim to minimize the power and maximize the admitted rate with the presence of constraints as follows:

$$\begin{aligned} \min_{\boldsymbol{X},\boldsymbol{M}} \quad & \varphi_{tot} && \text{(2a)} \\ \text{subject to} \quad & \sum_{s=1}^{S} \sum_{m_s=1}^{M_s} x_{m_s,n} \bar{\psi}_{\mathsf{C},s}^{\mathfrak{z},tot} \le \chi_{\mathsf{C},s}^n \ \ \forall n, && \text{(2b)} \\ & \sum_{s=1}^{S} \sum_{m_s=1}^{M_s} x_{m_s,n} \bar{\psi}_{\mathsf{S},s}^{\mathfrak{z},tot} \le \chi_{\mathsf{S},s}^n \ \ \forall n, && \text{(2c)} \\ & \sum_{s=1}^{S} \sum_{m_s=1}^{M_s} x_{m_s,n} \bar{\psi}_{\mathsf{M},s}^{\mathfrak{z},tot} \le \chi_{\mathsf{M},s}^n \ \ \forall n, && \text{(2d)} \\ & x_{m_s,n} \in \{0, 1\} \ \ \forall n, \forall s, \forall m_s && \text{(2e)} \\ & \bar{T}_s^{\ tot} \le T_{max}^s. && \text{(2f)} \end{aligned}$$

where, $\mathfrak{z} \in \{c, d\}$, and the constraints (2b), (2c), and (2d) specify that VNFs hosted by data center n cannot exceed the data center's total resources of CPU, memory, and the storage. Moreover, (2e), represents that the $x_{m_s,n}$ is a binary variable. In addition, (2f), indicates that the mean total delay of the system is less than the threshold. Moreover, $\boldsymbol{X}$ is the matrix of the $x_{m_s^{\mathfrak{z}},n}, \forall n, \forall m_s^{\mathfrak{z}}$ which defines the allocation of VNFs of slices to the resources of data centers. Furthermore, $\boldsymbol{M}$ is the vector of $M_s, \forall s$ that defines the number of pre-defined VNFs for each slice in the system.

B. Proposed Method

In the following, we present our approach for addressing the problem outlined in (2), which necessitates solving it across two distinct time scales. In the large time scale, we find the optimal number of pre-defined VNFs based on the mean arrival delay and the mean service time of the system at different times of network traffic. In the small time scale, we consider the problem of the dynamic service admission control based on the resource management of the VNFs in the system. Due to the dynamic nature of the small time-scale problem, we are able to solve it using deep reinforcement learning (DRL).

Therefore, firstly, we can simplify the constraint (2f) and find the sub-optimal value for the number of the VNFs in each slice s. Afterward, we use the DRL technique to solve the problem of the admission control system.

1) Estimation of The VNF Number: In this section, we want to find the optimal number of VNFs in the system for each slice in the large time scale based on the mean service time and mean arrival service rate.

In the lifecycle of the network slicing technique, we have four phases: Preparation, Commissioning, Operation and Decommissioning. The VNF numbers are estimated in the Preparation and Commissioning phase. However, in the Operation phase, it can be modified based on any change in the traffic of the system. However, with correct estimation, the power consumption of the system is reduced.

We can simplify and relax the constraint (2f). This constraint can be converted as $M_s \geq \frac{\bar{\alpha}_s}{\bar{\mu}_s - 2/T_{\text{max}}}$. Since we want to minimize the power consumption in the first term of the cost function of the problem (2), we consider the minimum value for the M_s, $\forall s$. As a result, since the number of VNFs is the integer, we have $M_s = \lceil \frac{\bar{\alpha}_s}{\bar{\mu}_s - 2/T_{\text{max}}} \rceil$.

2) Resource Management: This section introduces a DRL-based network slicing resource management for dynamic service admission control and power minimization in the small time scale after solving the sub-optimal number of VNFs. This process is done in the Commissioning and Operation phase of network slice life cycle.

In the O-RAN architecture, the DRL method is carried out in the xApp in the near RT RIC. The DRL approach combines deep neural networks (DNN) with reinforcement learning (RL). We use a DRL method to solve this problem since we have a dynamic system.

All RL techniques represent a Markov decision-making process with $(\mathcal{S}, A, R, P, \gamma)$. Firstly, A represents the action vector. $\mathcal{S}$ represents the state space matrix. Moreover, R_t is the accumulated reward function and r_t is a reward for taking action at time slot t. A probability of transfer is given by $P(.|\mathcal{S}, a)$. Furthermore, The discount factor is defined as $\gamma \in (0, 1]$. Moreover, the $\Pi(.|\mathcal{S})$ is the policy that maps the state to the distribution of actions. In addition, the value-state function for state $\int$ under the policy $\Pi(.|\mathcal{S})$ with $V^{\Pi}(\int)$ denotes the expected return value in state $\int$ under policy $\Pi(.|\mathcal{S}))$. Finally, The value of performing operation a in state $\int$ under the $\Pi(.|\mathcal{S})$ policy is shown as $Q^{\Pi}(\int, a)$.

In the RL method, the aim is to maximize the total reward specified as $R_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k}$.

3) Enviroment: This section introduces the Markov decision process (MDP) to describe an agent and environment based on the system model in Section II.

- State: The state is the position of the agents at a specific time. Assume in time step t, we have r_t^s request from slice s. In this problem, the state in time step t is $\mathbb{S}_t = \{\boldsymbol{\chi}_t, \boldsymbol{R}_t\}$. Where $\boldsymbol{\chi}_t \in \mathbb{C}^{N \times 3}$ is the 2D vector of remaining CPU, storage and memory for all data centers in time step t. Furthermore, $\boldsymbol{R}_t \in \mathbb{C}^S$ is the 1D vector of service requests in time step t.
- Action: The action in each time step t is represented by $\mathbb{A}_t = \{\boldsymbol{X}_t\}$, where $\boldsymbol{X}_t \in \mathbb{C}^{S \times N}$ is the 2D vector indicating whether the VNF of slice s is assigned to the data center n or not.
- Reward: The aim is to maximize the admission rate and to minimize the number of activated data centers. The reward in each time step t is defined by $\mathbb{R}_t$,
$$\mathbb{R}_t = \begin{cases} \varphi_{tot,t}, & \chi_{i,s}^n \geq 0 \;\; \forall n, i \in \{C, S, M\} \\ -M & \text{otherwise} \end{cases} \tag{3}$$
where $\varphi_{tot,t}$ is the cost function of the system in each time step t. Moreover, M is a large integer number.

As mentioned in Section III-A, in the problem 2, the action is a discrete binary vector, and the state is continuous. Hence, we use the proximal policy optimization (PPO) method to solve this problem.

4) PPO method: A Temporal Difference (TD) representation of the Policy gradient can be seen in the Actor-Critic model. In the actor-critic model, the system has two networks: the actor and the critic. Based on the actor's decision, the action is taken. The actor learns by applying the policy gradient method. The actor receives feedback on the correctness of the action from the critic network. The critic analyzes the actor using the value function. A PPO is a method based on actor-critic analysis [15].

The PPO algorithm is a policy gradient algorithm that balances simplicity, complexity, and tuning. By updating each step, it maintains a moderately low deviation from the previous policy. PPO is a reliable and efficient version of the trust region policy optimization (TRPO) algorithm that employs first-order optimization. Consequently, PPO combines actor-critic and TRPO concepts. It is critical to note that the TRPO technique ensures that the updated policy is not too different from the old policy. Hence, the updated policy is within the trust region of the old policy. The objective function of TRPO can be formulated as follows.

$$\max_{\theta} \quad \hat{\mathbb{E}}_t[\frac{\pi_\theta(a_t|\int_t)}{\pi_{\theta_{old}}(a_t|\int_t)}\hat{A}_t] \tag{4a}$$

$$\text{subject to} \quad \hat{\mathbb{E}}_t[\text{KL}[\pi_{\theta_{old}}(.|\int_t), \pi_\theta(a_t|\int_t)]] \leq \delta, \tag{4b}$$

where π_θ is a stochastic policy and $\pi_{\theta_{old}}$ is the policy vector before updating. Moreover, $\hat{\mathbb{E}}_t[.]$ is the average of several samples and $\hat{A}_t$ is the advantage function estimator in the time of t. In the TRPO method, in order to enable the trust region for optimization, KL divergence constraints must be met. By modifying the clipped substitute objective function, PPO applies the policy constraint. Assume $r_t(\theta) = \frac{\pi_\theta(a_t|\int_t)}{\pi_{\theta_{old}}(a_t|\int_t)}$. In the PPO method the main objective function is $L^{CLIP}(\theta) = \hat{\mathbb{E}}_t[\min(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t)]$. where $1-\epsilon$ and $1+\epsilon$ are the lower and upper clipping ranges for state action $(\int, a)$. Moreover, the $r_t(\theta)$ is clipped to the lower and upper bound if it is out of this range [16].

IV. Secure MTD Service Admission Control

This section aims to secure the proposed energy-efficient service admission control against adversarial attacks. Firstly, we introduce the feasible attack model considered in this study. Then, we present the MTD strategy proposed to secure the system. As depicted in Fig. 1, the DRL-based service admission control service can be deployed as a xApp in the Near-RT RIC. To apply the MTD technique, the system requires different trained models. Each model is learned and deployed in a specific xApp. Therefore, we have different xApps consisting of DRL methods with similar performance.

A. Adversarial Attack Model

We describe a feasible malicious attack on the proposed DRL method. There are three types of attacking the ML system based on the attacker's knowledge of the targeted model (i.e., model's parameters and architecture) and training data. The adversarial attack is considered white-box, gray-box, or black-box when the attacker obtains full, partial, or no knowledge, respectively [8]. Here, we assume that the adversary is targeting the PPO models under a black-box setting, i.e., no knowledge on the targeted model. To attack the system, we apply a weak adversary attack based on [17]. Assume that attackers want to attack the system at time t. The attackers generate an arbitrary state $\hat{s}_t$ and the corresponding reward function $\hat{r}(\hat{s}_t, .)$. After the agent determines the altered state $\hat{s}_t$, it carries out the action a_t and observes $\hat{r}(\hat{s}_t, a_t)$, instead of $r(s_t, a_t)$.

B. Moving Target Defense Strategy

MTD, an emerging security strategy, continuously alters system configurations, making attacks challenging due to increased uncertainty and complexity. This method lowers attack success rates by reducing attacker knowledge and effectiveness. MTD enhances defense by adding ambiguity and offering multiple configurations [11].

We deploy four diverse PPO models with varied configurations into different xApps in this scenario. These four methods give us almost similar results but with different configurations in terms of the number of neural network layers, batch size, discount factor, learning rates, among others.It is critical to note that attackers have a set of attacks that are designed to attack the configurations of the defender. In the process of training, the attacker randomly chooses one of the xApps that contains one of these PPO models and attacks it.Hence, once these four models have been trained, one random model is chosen from the four to run on each input and return the output generated by that model. As attacks are directed at one of the models, attackers have less impact on the system since they do not know which model the system selects at a given time. Therefore, the probability of an adversarial attack against our system is diminishing by randomly choosing one of these models that can be the un-attacked model.

V. Numerical results

This section presents numerical results for the main problem. Considering the similarity of packets between O-CU and O-DU, their requirements are equivalent. Only minor headers in O-CU packets are removed in O-DU, having negligible impact on processing. Therefore, we assume O-CU and O-DU share the same processors (VNFs). In these figures, we consider two data centers, each equipped with a CPU boasting 32 cores, 50GB memory, and 5TB storage.

Assume there are two service requests. Each service is assigned to a specific slice. Each slice contains M_s, pre-defined similar VNFs that are obtained from the large time scale. For the first service, each request needs 2 cores, 7GB memory, and 30GB storage in O-CU and O-DU. The second service requires 3 cores, 5GB memory, and 50GB storage per request in O-CU and O-DU.

To assess the performance of the proposed solutions, we illustrate five different scenarios. The first scenario is the exhaustive search. The system works with the PPO model in the second scenario without attack. The third scenario involves an adversarial attack on the system without protection. In the fourth scenario, the protected MTD system is under attack. The fifth scenario considers the baseline method, in which random allocation is assumed. The training process involves learning four PPO models with different parameters (different batch sizes, discount factor, learning rate, the number of steps to run for each environment, etc.). The models have similar performance. At each time slot, one of these models is chosen randomly to protect the system from attack.

Fig. 2 displays the mean reward over time slots for a system without service admission control attacks. This system features 12 service arrival rates per time slot for two distinct services with varying QoS, demonstrating PPO convergence. In Fig. 3, the service admission rate is depicted for different service arrival rates for the five scenarios. In this simulation, we assume that the design factor κ is large enough that the cost function is affected just by the admission control system, and the power consumption is not considered here. The figure shows a dynamic service arrival in a system. Each time slot has a 30% service departure rate. This figure indicates that over 80% of admissions were recorded whenever we had the average of six service arrival rates for each service in each time slot in the system without any attack. As service arrival rates rise, admission rates fall due to increased packet arrivals and traffic, leading to reduced admission rates at the system's fixed capacity The system's performance decreases by at most 93% when under attack, showing the considerable impact adversarial attacks can have on unprotected ML models. The MTD-protected system has significantly improved the system's robustness, yielding a 70% increase in service admission rate compared to the attacked system. Moreover, the system's performance increases 62% compared to the baseline, which is the random allocation. The optimality of the PPO model decreases to 16.7% as the service arrival rate increases to 12.

In Fig. 4, we present the normalized power consumption across five scenarios, varying with service arrival rates. The parameter κ relates power consumption to admission control, albeit power usage remains substantial relative to admission control. Nevertheless, admission control's impact is non-negligible. In our simulation, baseband processing power $\phi_{m_s,n}$ is uniformly distributed within the range [100,200]. The

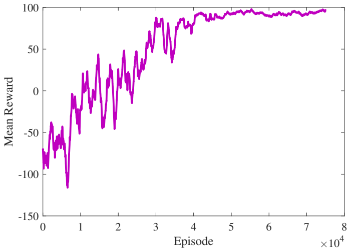

Fig. 2 Mean Reward vs. Time Slots.

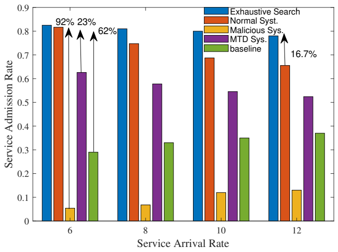

Fig. 3 Service Admission Rate vs. Service Arrival Rate.

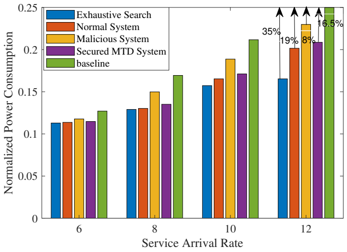

Fig. 4 Power Consumption vs. Service Arrival Rat.e

TABLE I: un-estimated VNF numbers vs. the extra power consumption

Number of VNFs	Extra Power Consumption
12	27 %
10	21 %

figure highlights that higher service arrival rates correspond to increased system power consumption. The malicious system consumes more power due to the inverse power-reward relationship. Despite denied control requests from attacks, the attacker aims to maximize power, vital for DRL rewards. The power of the normally trained system is decreased 19% compared to the baseline system, which is the random allocation. The power of the PPO system is increased 11% (based on the baseline method) after the system is attacked. Whenever we apply the MTD technique to an attacked system, the power is reduced by 8.5%. Also, the power consumption of the PPO system increased 16% (based on the baseline method) compared to the optimal method. Assume the delay of the service in the system is $T_s = 1.07\mu sec$. The mean service rate is considered to be 2 Mbps and the mean arrival rate of the system is 1 Mbps. The estimated number of VNFs is 8. In Table I, the extra power consumption ratio for 10 and 12 VNFs is obtained. If we apply 10 VNFs instead of 8 VNFs, the mean power increases by 21%. Considering 12 VNFs instead of 8 VNFs, the mean power increased 27%. In the network slicing life cycle (as in III-B1), optimal VNF estimation happens during preparation and commissioning, the planning phase. However, adjustments are possible during operation, reducing excess power use caused by inaccurate initial estimates

VI. Conclusion

In this academic study, we address two pivotal challenges within the O-RAN architecture: firstly, the determination of the optimal number of VNFs for each slice, and secondly, the establishment of secure AI/ML methodologies for the dynamic management of service admission control and power reduction within the O-RAN framework. We derive sub-optimal VNF quantities at a larger time scale and employ an actor-critic approach with the PPO algorithm at a smaller scale. Four PPO models are trained in distinct xApps within the near RT RIC. Security is bolstered using MTD, randomly selecting xApps with trained PPO models for added unpredictability. Numerical results demonstrate robust PPO performance in the absence of attacks, significantly improved by the MTD strategy in adversarial scenarios. To enhance system security with MTD, we need to train multiple models, despite its limitations. Future MTD techniques should aim for a balance between robustness, performance, and cost.

Acknowledgment

This research work is partially supported by the Business Finland 6Bridge 6Core project under Grant No. 8410/31/2022, the Research Council of Finland (former Academy of Finland) IDEA-MILL project (Grant No. 352428), the Research Council of Finland (former Academy of Finland) 6G Flagship program (Grant No. 346208), and the European Union's Horizon Europe research and innovation programme HORIZON-JU-SNS-2022 under the RIGOUROUS project (Grant No. 101095933) and the 6G-SANDBOX project (Grant No. 101096328). The paper reflects only the authors' views. The Commission disclaims responsibility for any use of the information it contains.

References

[1] S. Kukliński *et al.*, "6G-LEGO: A Framework for 6G Network Slices," *Journal of Commun. and Networks*, vol. 23, no. 6, pp. 442 – 453, 2021.

[2] W. Wu *et al.*, "AI-native Network Slicing for 6G Networks," *IEEE Wireless Communications*, vol. 29, no. 1, pp. 96–103, 2022.

[3] M. K. Motalleb *et al.*, "Resource Allocation in Open RAN System using Network Slicing," *IEEE Trans. Net. Service Manag.*, pp. 1–1, 2022.

[4] C. Benzaïd *et al.*, "AI-based Autonomic & Scalable Security Management Architecture for Secure Network Slicing in 5G," *IEEE Network*, 2022.

[5] O-RAN Alliance, "O-RAN Slicing Architecture 8.0," in *Technical Specification*, Oct. 2022.

[6] H. Zhang *et al.*, "Federated Deep Reinforcement Learning for Resource Allocation in O-RAN Slicing," *arXiv preprint:2208.01736*, 2022.

[7] Q.-T. Luu *et al.*, "Admission Control and Resource Reservation for Prioritized Slice Requests with Guaranteed SLA under Uncertainties," *IEEE Trans. Net. Service Manag.*, 2022.

[8] C. Benzaïd and T. Taleb, "AI for Beyond 5G Networks: A Cyber-SecurityDefense or Offense Enabler?" *IEEE Network*, vol. 34, no. 6, pp. 140 – 147, Nov./Dec. 2020.

[9] ——, "AI-driven Zero Touch Network and Service Management in 5G and Beyond: Challenges and Research Directions," *IEEE Network*, vol. 34, no. 2, pp. 186 – 194, Mar/Apr. 2020.

[10] ——, "ZSM Security: Threat Surface and Best Practices," *IEEE Network*, vol. 34, no. 3, pp. 124 – 133, May/June 2020.

[11] S. Sengupta *et al.*, "MTDeep: Boosting the Security of Deep Neural Nets against Adversarial Attacks with Moving Target Defense," in *Decision and Game Theory for Security, Springer-Verlag*, 2019, pp. 479 –491.

[12] A. Rashid and J. M. Such, "StratDef:A Strategic Defense against Adversarial Attacks in Malware Detection," *ArXiv*, 2022.

[13] Y. Qiu, J. Wu *et al.*, "MT-MTD: Muti-Training based Moving Target Defense Trojaning Attack in Edged-AI network," in *ICC-IEEE International Conference on Communications*. IEEE, 2021, pp. 1–6.

[14] M. Karbalaee *et al.*, "Joint Power Allocation and Network Slicing in an open RAN System," *arXiv preprint arXiv:1911.01904*, 2019.

[15] S. Mollahasani *et al.*, "Dynamic CU-DU Selection for Resource Allocation in O-RAN Using Actor-Critic Learning," in *2021 IEEE GLOBECOM*, 2021, pp. 1–6.

[16] Y. Wang *et al.*, "Trust Region-guided Proximal Policy Optimization," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[17] T. Wu *et al.*, "On reinforcement learning with adversarial corruption and its application to block MDP," in *International Conference on Machine Learning*. PMLR, 2021, pp. 11 296–11 306.