Ataque Adversarial de Evasão a Sistema de Detecção de Intrusão e Métodos de Defesa em Redes Open RAN

Victor Dias[1], Murilo Silva[2], Matheus Gomes[1], Lucas Borges[1 2], André Riker[1], Antônio Abelém[1]

[1]Programa de Pós-Graduação em Ciência da Computação (PPGCC)
Universidade Federal do Pará (UFPA)

[2]Rede Nacional de Ensino e Pesquisa (RNP)

victor.leite@ig.ufpa.br, matheus.cordovil@itec.ufpa.br
{murilo.silva, lucas.oliveira}@rnp.br
{ariker, abelem}@ufpa.br

Abstract. *Intrusion Detection Systems (IDS) are an important element for open radio access networks (Open RAN), as they provide essential defense services against cyber threats. Typically, IDS in Open RAN relies on machine learning models, which makes it vulnerable to adversaries aiming to evade detection. In particular, a DDoS attack capable of evading the IDS of an Open RAN infrastructure poses a significant risk to the telecommunications service provider. Nevertheless, few studies have focused on adversarial attacks targeting IDS in Open RAN networks. In this context, this paper presents an adversarial attack called Distributed DoS Adversarial Detection Evasion (DDoS-ADE), which is designed to evade DDoS detection by an IDS deployed in the RAN Intelligent Controller (RIC) of an Open RAN infrastructure. In addition to presenting this evasion attack, the paper investigates the effectiveness of two defense strategies: (i) model feature reduction and (ii) adversarial training. Both the proposed DDoS-ADE attack and the defense methods were implemented and evaluated in the OpenRAN@Brasil testbed. The results show that DDoS-ADE is able to evade 94,66% of an unprotected IDS. In comparison, the implemented defense methods are capable of reducing evasion by up to 98,45%.*

Resumo. *Os Sistemas de Detecção de Intrusão (IDS) são elementos importantes para redes de acesso via rádio abertas (Open RAN), pois fornecem serviços essenciais de defesa contra ameaças cibernéticas. Tipicamente, os IDS em ambientes Open RAN dependem de modelos de aprendizado de máquina, o que os torna vulneráveis a adversários que buscam escapar da detecção. Em especial, um ataque DDoS capaz de contornar o IDS de uma infraestrutura Open RAN representa um risco significativo para o provedor de serviços de telecomunicações. No entanto, poucos estudos têm se dedicado a ataques adversariais direcionados a IDS em redes Open RAN. Nesse contexto, este trabalho apresenta um ataque adversarial denominado Distributed DoS Adversarial Detection Evasion (DDoS-ADE), projetado para evadir a detecção de ataques DDoS por um IDS implantado no Controlador Inteligente de RAN (RIC) de uma infraestrutura Open RAN. Além de apresentar esse ataque de evasão, o artigo investiga a eficácia de duas estratégias de defesa: (i) redução das características do modelo e (ii) treinamento adversarial. Tanto o ataque DDoS-ADE*

quanto os métodos de defesa foram implementados e avaliados no testbed OpenRAN@Brasil. Os resultados demonstram que o DDoS-ADE é capaz de evadir 94,66% das detecções de um IDS não protegido. Em comparação, os métodos de defesa implementados conseguem reduzir a evasão em até 98,45%.

1. Introdução

As redes móveis têm evoluído e adotado um novo paradigma de infraestrutura para comunicações móveis chamado *Open Radio Access Network* (Open RAN) [Marinova and Leon-Garcia 2024] ou redes de acesso via rádio aberta. Esta abordagem inovadora permite diminuição de custos e leva a *softwerização* da rede [O-RAN Alliance 2018]. Apesar das vantagens em termos de redução de custos, a segurança é uma questão crítica neste ambiente [Polese et al. 2023].

Umas das inovações viabilizadas pelo Open RAN são as xApps (*eXtended Applications*). Estas aplicações são executadas no controlador inteligente de RAN e funcionam com uma latência de quase tempo real, por isso as xApps são implementadas no *Near-Real Time RAN Intelligent Controller* (Near-RT RIC). Do ponto de vista da cibersegurança, as xApps podem incorporar modelos de Aprendizado de Máquina (AM) para detectar padrões de tráfego maliciosos. Isto permite que xApps atuem como Sistema de Detecção de Intrusão (IDS) - *Intrusion Detection System* (IDS) para ataques DDoS.

Existem diferentes formas de implementar um IDS em Open RAN, incluindo abordagens baseadas em heurísticas, com a escolha definida em função das características do problema [Amachaghi et al. 2024]. Contudo, no contexto do Open RAN, há uma ênfase crescente na adoção de soluções fundamentadas em IA/AM, pois estas são elementos centrais de sua arquitetura, oferecendo benefícios como automação em redes *zero-touch*, otimização contínua e maior eficácia na detecção de ameaças.

No entanto, uma xApp que implementa IDS, baseada em AM, possui vulnerabilidades conhecidas que podem ser exploradas por meio dos denominados ataques adversariais. Há diversos tipos de ataques adversariais, destacam-se os de evasão e extração. Estes ataques são capazes de fazer um modelo de AM gerar uma resposta errada ou extrair informações sensíveis do modelo. No contexto de Open RAN, existe o risco de um atacante lançar um ataque adversarial e evadir a detecção da xApp IDS. Os efeitos dessa evasão são de grande dano aos usuários da rede e também à operadora [Ayub et al. 2020].

Diante dessa vulnerabilidade, neste trabalho detalhamos um ataque adversarial, denominado *Distributed DoS Adversarial Detection Evasion* (DDoS-ADE), com o objetivo de burlar a detecção DDoS realizada por uma xApp IDS executada no RIC da arquitetura Open RAN. Adicionalmente, apresentamos estratégias de defesa fundamentadas em (i) treinamento adversarial e (ii) diminuição do número de características do modelo, e analisamos a eficácia de cada estratégia, por meio de métricas fundamentais como o Recall. Tanto o ataque DDoS-ADE quanto às estratégias de defesa foram implementadas e avaliadas em um ambiente real, provido pelo testbed do projeto OpenRAN@Brasil.

O restante deste trabalho está estruturado da seguinte forma: A Seção 2 apresenta o referencial teórico. A Seção 3 discute os trabalhos relacionados. As Seções 4 e 5 apresentam o ataque proposto e os métodos de defesa. A Seção 7 detalha a avaliação e resultados obtidos. A Seção 8 apresenta a conclusão e trabalhos futuros.

2. Referencial Teórico

No contexto de segurança em Open RAN, é fundamental compreender conceitos essenciais de sua arquitetura e os seus desafios. Desse modo, nesta seção serão abordados os temas relevantes para compreensão deste trabalho, incluindo a visão geral da arquitetura Open RAN, o aprendizado de máquina em Open RAN e os ataques de evasão.

2.1. Redes Open RAN

O conceito Open RAN traz a desagregação de hardware e software, permitindo múltiplos fornecedores de equipamentos e a transição para tecnologia baseada em nuvem. Esta abordagem aumenta a flexibilidade, a interoperabilidade e promove a inovação nas redes de próxima geração, sendo necessário que haja padronização dos elementos da arquitetura Open RAN. A O-RAN Alliance e o 3GPP (3rd Generation Partnership Project) atuam na padronização dos componentes e interfaces dessa arquitetura [Polese et al. 2023].

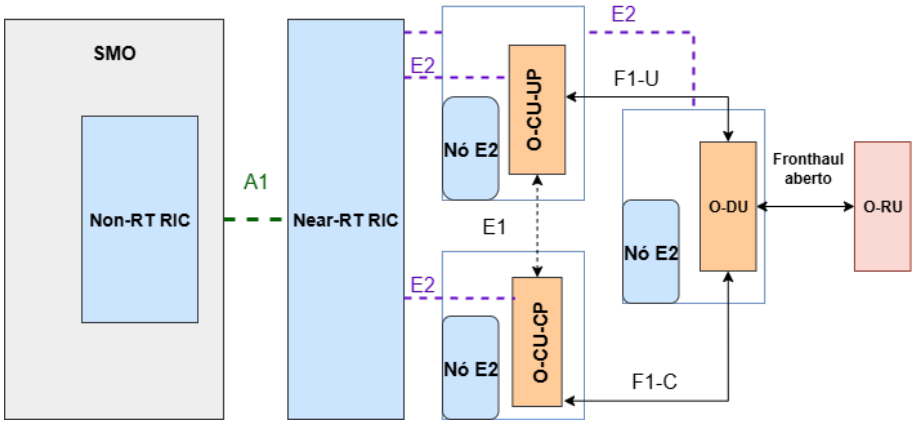

Figura 1. Arquitetura Open RAN (Adaptada [Marinova and Leon-Garcia 2024]).

Conforme ilustrado na Figura 1, entre os principais nós lógicos arquitetura Open RAN, a O-CU (Unidade Central Open RAN) desempenha funções de plano de controle (CP), é responsável por hospedar o controle de recursos de rádio e uma porção de controle do protocolo convergência de dados em pacotes. Além disso, a O-CU também é encarregada de executar as funções do plano de usuário (UP). A O-DU (Unidade Distribuída) implementa as camadas inferiores da pilha de protocolos, especificamente as camadas controle de enlace de rádio, atuando no processamento de dados em tempo real e na interface direta com as unidades de rádio. Por fim, a O-RU (Unidade de Rádio) é dedicada à execução da Low-PHY (Camada Física Inferior) e ao processamento dos sinais de radiofrequência, estabelecendo a interface com o ambiente físico de transmissão e recepção de sinais.

A parte inteligente dessa arquitetura é designada aos Controladores Inteligente da RAN (RIC): O Non-RT RIC (Controlador Inteligente de RAN em Tempo Não Real) e o Near-RT RIC (Controlador Inteligente de RAN em Tempo Quase Real). O Non-RT RIC é responsável por otimizar a RAN via interface A1, utilizando modelos de aprendizado de máquina e inteligência artificial para treinamentos, atualizações e orientações baseadas em políticas, operando em ciclos com mais de 1s. Por outro lado, o Near-RT RIC permite o controle e a otimização quase em tempo real dos serviços e recursos da RAN, operando em ciclos de aproximadamente 10ms a 1s, com base em dados obtidos pela interface E2. Este controlador hospeda xApps, desenvolvidas como microsserviços independentes, que

utilizam a interface E2 e possibilitam inovação contínua no ambiente da RAN por meio da coleta e análise de dados em tempo quase real.

A comunicação entre os componentes é feita por meio de interfaces padronizadas. A interface E2 conecta os principais nós da RAN ao Near-RT RIC, permitindo a aplicação de políticas de controle dinâmico e a coleta de informações operacionais em ciclos de baixa latência. Outra interface relacionada ao Near-RT RIC é a A1, que o conecta ao Non-RT RIC e permite troca de políticas de otimização e de configurações operacionais.

2.2. Aprendizado de Máquina em Open RAN

No contexto de Open RAN, o uso do Aprendizado de Máquina (AM) é inserido nos RICs, devido à pluralidade de dados disponíveis. Algoritmos de IA/AM executados no RIC podem ser utilizados nas mais variadas aplicações, como: segurança da rede, detecção de anomalias e previsão de utilização de recursos. Os dados são coletados pelas interfaces Open RAN (O1, E2 ou A1). Além disso, esses dados também podem ser coletados de UE (Dispositivo do Usuário) ou Core da Rede [Alliance 2021]. Esses dados são utilizados por funções de treinamento e inferência de aprendizado de máquina, conhecido como *host* de treinamento de AM e *host* de inferência de AM [Polese et al. 2023]. O *host* de treinamento de AM hospeda o treinamento online e offline do modelo, normalmente, no Non-RT RIC, o Neart-RT RIC pode ser utilizado em alguns cenários. O *host* de inferência de AM hospeda o modelo durante a inferência para a execução do modelo de aprendizado online, ambos os RICs podem ser utilizados, esta arquitetura é ilustrada na Figura 2.

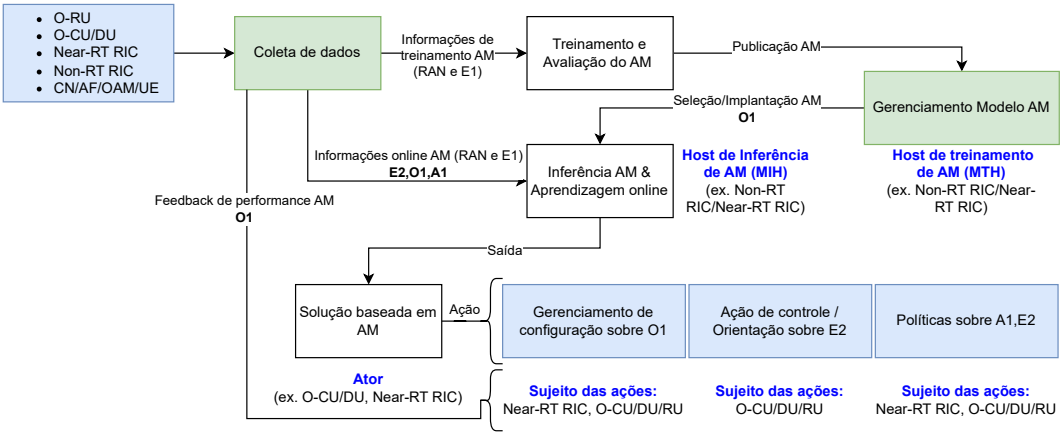

Figura 2. Arquitetura *Framework* AM Open RAN (Adaptada [Rimedo Labs 2023]).

O *host* de inferência de AM gera resultados utilizados por componentes, como O-CU/DU e RICs. Com base no resultado da inferência eles podem tomar ações dependendo da sua função, realizando mudanças de configurações, aplicando novas políticas ou controlando dinamicamente determinados parâmetros. Após a implementação dessas ações, os componentes geram novos dados, que alimentam o próximo ciclo de inferência para melhorar seu desempenho baseado no *feedback* recebido. Desse modo, essas decisões podem resultar em diferentes ações, incluindo o ajuste de configurações via interface O1, gerenciamento de políticas A1 ou modificações no controle e nas políticas via E2.

2.3. Ataques de Evasão

Para entender o ataque de evasão primeiro é necessário entender o conceito de Aprendizado de Máquina Adversarial (Adversarial Machine Learning), é um campo de estudo que

teve seu início com base no artigo de Szegedy et. al [Szegedy et al. 2014]. Esse trabalho introduziu o conceito de exemplo adversarial, que consiste na modificação de uma entrada de dados de forma sutil, mas o suficiente para impactar de forma negativa o modelo de aprendizado de máquina utilizado, induzindo erros em seu processo de inferência tanto para problemas de regressão quanto para classificação. Vale destacar que o ataque de evasão ocorre sem influenciar os dados de treinamento. O grande diferencial dessa abordagem estava na adoção de métodos de otimização que são utilizados no aprendizado de máquina. Desde então, uma série de estudos se dedicaram, e ainda se dedicam, a mapear esses tipos de ataque e como eles podem ser evitados ou remediados [Costa et al. 2024].

A respeito do nível de conhecimento do atacante sobre o modelo, os ataques podem ser classificados como de caixa branca (*white-box*) ou caixa preta (*black-box*). No ataque de caixa branca, o adversário tem acesso completo ao modelo, incluindo sua arquitetura, dados de treinamento e parâmetros, o que permite explorar vulnerabilidades com maior precisão por meio de exemplos adversariais. No ataque caixa preta, o atacante não conhece detalhes internos do modelo e baseia suas ações apenas em observações das entradas e saídas [Nicolae et al. 2019a]. Os objetivos desses ataques variam, podendo incluir a redução da confiabilidade do sistema, classificações incorretas sendo elas aleatórias ou direcionadas e a manipulação da saída para uma classe específica.

3. Trabalhos Relacionados

O cenário envolvendo IDS em Open RAN ainda é recente, entretanto identificam-se trabalhos que seguem essa linha de pesquisa como o de [Chang et al. 2024] que desenvolve um sistema de detecção de intrusão baseado em aprendizado profundo para o *Open Fronthaul* da arquitetura Open RAN com o objetivo de detectar ataques DDoS, utilizando diversos modelos de aprendizado profundo e os *datasets* CICIDS2017 [Sharafaldin et al. 2018] e CICDDoS2019 [Sharafaldin et al. 2019]. Porém, a solução [Chang et al. 2024] tem como fator limitante sua implementação, embora o texto afirme que é um IDS para arquitetura Open RAN, o trabalho limitou-se a validar os modelos de aprendizado profundo com base em *datasets* fora do contexto Open RAN. Além disso, ataques adversariais não foram abordados na proposta, o que abre margem para uma vulnerabilidade crítica no sistema.

Contudo, se tem registro da utilização de xApp em segurança em Open RAN, conforme é visto no trabalho de [Xavier et al. 2023], que utiliza uma versão própria do Near-RT RIC e um *framework* de classificação adaptado para uma xApp que utiliza aprendizado de máquina para detecção de ataques DoS e DDoS. Ademais, traz uma discussão interessante sobre o tempo de resposta dos modelos de AM utilizados, esse é um ponto fundamental, devido o Near-RT RIC exigir operações sensíveis ao tempo. A solução apresenta resultados relevantes, além da análise da eficácia da solução, ao utilizar uma xApp alguns requisitos de performance são levados em consideração, o principal sendo seu tempo de resposta. Desse modo, o trabalho foi competente em avaliar os impactos dos diferentes modelos de AM utilizados e como eles podem impactar no tempo de ação da solução. Entretanto, o estudo não considerou que os modelos de AM podem ser atacados.

Quanto aos ataques adversariais de evasão, o trabalho de [Ergu et al. 2025] explora vulnerabilidades de segurança em redes Open RAN para comunicação veículo-para-tudo, focado em ataques adversariais contra a alocação de recursos baseada em aprendizado por reforço profundo. O ataque implementado manipula observações ambientais para enganar o modelo utilizado, resultando em alocações incorretas e redução nas taxas de transmissão para veículos. Os resultados do trabalho demonstraram uma significativa

queda nas taxas de dados dos usuários e nas taxas de entrega de pacotes.

Por outro lado, o trabalho de [Sapavath et al. 2023] demonstra como um ataque adversarial realizado por uma xApp maliciosa infiltrada pode comprometer a precisão de uma xApp legítima, que tem como função a classificação de interferência, baseada em uma rede neural profunda. O ataque implementado levou a inferências incorretas pela aplicação legítima e consequentemente à degradação da rede devido a sua tomada de decisões. Embora os trabalhos de [Ergu et al. 2025] e [Sapavath et al. 2023] considerem a ocorrência de ataques nos modelos de AM utilizados em suas soluções, demonstrando como diferentes tipos de ataques de evasão podem comprometer a eficiência de uma xApp, eles não implementam medidas para mitigar esses ataques.

Em suma, os trabalhos apresentados expõem resultados relevantes abordando o impacto de ataques DDoS em estruturas sensíveis das redes Open RAN e destacam a importância de soluções que usem ferramentas nativas dessa arquitetura, como as xApps. No entanto, uma limitação observada é a vulnerabilidade de modelos de aprendizado de máquina a ataques de evasão, dificultando a detecção e aumentando a degradação da rede. Deste modo, este trabalho propõe a implementação de métodos de defesa adversarial para a mitigação do impacto de ataques de evasão em um IDS focada em ataques DDoS no cenário Open RAN. Bem como, uma discussão do impacto da adoção desses métodos, como tempo de treinamento, métricas de desempenho do modelo e sua eficácia. Podem-se destacar duas principais contribuições deste artigo: (i) a proposta de um ataque de evasão contra um IDS em redes Open RAN, utilizando técnicas como FGSM (Método do Sinal do Gradiente Rápido) e PGD (Descida do Gradiente Projetada); e (ii) a análise da eficácia de métodos de defesa adversarial frente a esses ataques.

4. O Ataque Adversarial de Evasão: DDoS-ADE

Esta seção apresenta o ataque que visa evadir um IDS quanto à detecção de um ataque DDoS em Open RAN, denominado *Distributed DoS Adversarial Detection Evasion* (DDoS-ADE). Para isso, essa seção apresenta em 4.1 uma visão geral do ataque em uma rede Open RAN, em 4.2 há a apresentação do modelo de ameaça considerado e em 4.3 é detalhado a execução do ataque.

4.1. Visão Geral

Na arquitetura Open RAN, dada a sua grande aplicação de modelos IA/AM nos RICs, um agente malicioso pode adulterar a entrada de dados usados como entrada de modelos IA/AM durante a fase de inferência no Near-RT RIC ou Non-RT RIC (modelo, parâmetros do modelo, entre outros). Essas ações impactam diretamente um dos três pilares da segurança da informação, a integridade [Alliance 2025b]. O relatório de segurança mais recente da O-RAN Alliance [Alliance 2025a] detalha riscos de presença de aplicações maliciosas em sua arquitetura. Além disso, alguns trabalhos [Sapavath et al. 2023] detalham como manipular os dados de outras aplicações comprometendo seu funcionamento.

A Figura 3 ilustra um cenário onde UEs executam um ataque DDoS direcionado à infraestrutura Open RAN, e um agente malicioso, uma xApp comprometida, é executado no ambiente de controle da RAN (RICs). Essa xApp atua como um agente interno com acesso legítimo à infraestrutura do Open RAN, mas que opera de forma ativa e maliciosa, comprometendo o processo de inferência da aplicação de detecção de ataques DDoS.

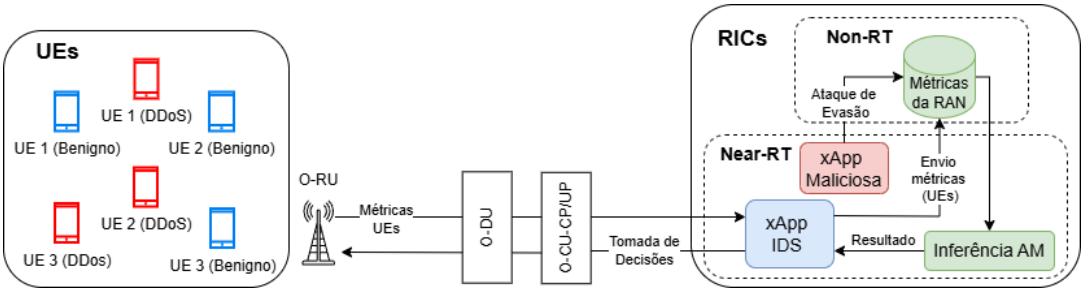

Figura 3. Ataque Adversarial de Evasão DDoS para xApp IDS em Open RAN.

4.2. Modelo de Ameaça

A fim de delimitar o escopo deste trabalho, definimos a seguir o modelo de ameaça do ataque de evasão considerado. Assume-se que há em execução no Near-RT RIC Open RAN uma xApp maliciosa com acesso *white-box*, ou seja, conhece a arquitetura e parâmetros do modelo alvo. Além disso, existem UEs maliciosos executando um ataque DDoS. O modelo de IA/ML alvo do ataque de evasão é executado em uma xApp IDS que monitora a base de dados contendo métricas coletadas das UEs para detecção de ataques DDoS e outros. Essa base de dados armazena as métricas coletadas da UEs, tais como: latência, transferência e perdas de pacotes.

A xApp maliciosa executa o ataque adversarial de evasão promovendo a alteração dos dados originais armazenados na base de dados localizada no RIC Open RAN. Tal alteração é possível, pois, durante a integração dos xApps, *malwares* podem ser instalados por um adversário nos xApps, obtendo assim acesso não autorizado ao Near-RT RIC. Isso ocorre por meio da exploração de mecanismos de autenticação fracos ou mal configurados nesse ambiente. O adversário pode, por exemplo, criar uma imagem de xApp malicioso e instalá-la durante a fase de integração. Além disso, um xApp legítimo pode ser clonado e inserido no Near-RT RIC por um agente interno malicioso [Alliance 2025a]. O objetivo do ataque é fazer o modelo alvo classificar de forma incorreta os tráfegos maliciosos gerados por UEs comprometidas. Dessa forma, o tráfego ofensivo permanece invisível ao IDS, permitindo que o ataque DoS ou DDoS continue sem ser detectado.

4.3. Especificação do Ataque

Dessa forma, este trabalho aborda como cenário um ataque adversarial de evasão que é executado por uma xApp maliciosa que tem como intuito evadir a detecção DDoS de um IDS xApp. A Figura 4 mostra este cenário, onde UEs se conectam a rede Open RAN e começam a realizar ataques DDoS. Após o xApp (IDS) responsável pela coleta de métricas enviar as métricas para o módulo de AM responsável pela inferência, esses dados podem ser alterados para um padrão adversarial, com o intuito de evadir a detecção do ataque. Dessa forma, o UE permanece na rede e consequentemente o ataque DDoS não é detectado.

Embora grande parte dos estudos sobre ataques adversariais seja voltada para modelos de classificação de imagens, onde pequenas perturbações são de grande eficácia gerando impactos significativos no desempenho do modelo, trabalhos como de [Ergu et al. 2025] demonstram a manipulação de métricas de rede por meio de agentes maliciosos, para gerar exemplos adversariais baseados em gradiente. Esses tipos de ataque adversarial consiste na utilização das informações de gradiente do modelo da vítima

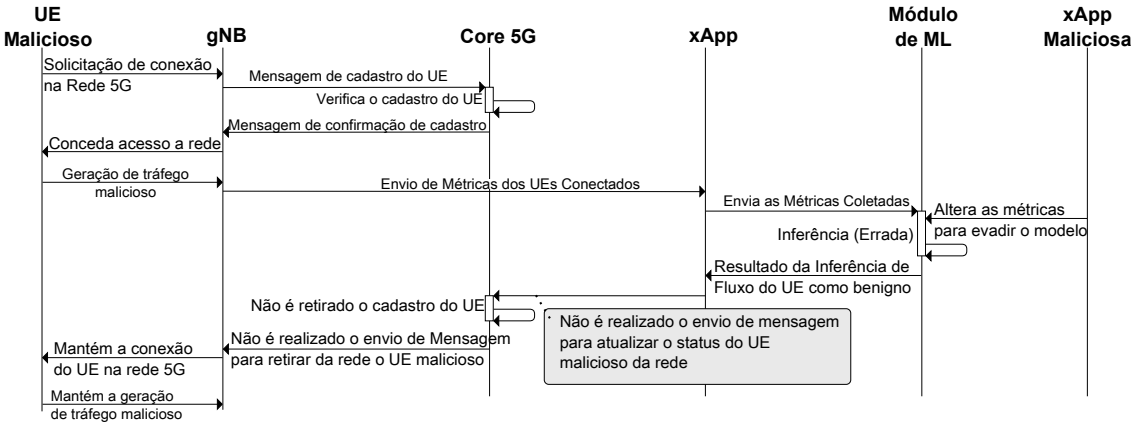

Figura 4. Diagrama de Sequência do IDS durante um ataque de Evasão.

para gerar os exemplos adversariais. Com isso, foram definidos os seguintes métodos para o ataque de evasão neste trabalho: i) FGSM e o ii) PGD.

O FGSM foi desenvolvido originalmente para atacar redes neurais profundas e possui como princípio maximizar uma determinada função de perda $J(x; w)$, que está sujeita a um limite na perturbação, por exemplo, $\|x - x_0\|_\infty \leq \varepsilon$. A função de perda pode ser interpretada como a perda gerada ao se avaliar um ponto de dados x em um classificador parametrizado por w. Desse modo, para evadir a detecção do modelo, deve-se maximizar essa perda. A Equação 1 é utilizada para obter o exemplo adversarial.

$$x = x_0 + \varepsilon \cdot \text{sign}(\nabla_x J(x_0; w)) \tag{1}$$

O ataque PGD é uma forma mais sofisticada do ataque FGSM que atua performando múltiplas iterações de perturbação de forma gradual para aumentar a eficácia do ataque adversarial. Ao invés de realizar uma única perturbação na direção do gradiente, o PGD realiza múltiplas iterações, limitando a magnitude da sua perturbação em direção ao gradiente a cada iteração, essa característica permite a criação de exemplos adversariais mais robustos e de difícil detecção.

$$x_{t+1} = \Pi_{\mathcal{B}_\varepsilon(x_0)} \left(x_t + \alpha \cdot \text{sign} \left(\nabla_x J(x_t, y) \right) \right) \tag{2}$$

A Equação 2 é utilizada para gerar o exemplo adversarial, as alterações são projetadas de forma a permanece dentro de um limite especificado em torno do estado original, x_t é o estado da iteração t. O ataque PGD atualiza o estado x_t para x_{t+1} e Π indica o operador de projeção sobre o conjunto $\mathcal{B}_\varepsilon(x_0) = \{x : \|x - x_0\|_\infty \leq \varepsilon\}$ x_0.

5. Métodos de Defesa Considerados

Existem diversos métodos para mitigar ataques de evasão, muitos deles presentes no repositório da ferramenta *Adversarial Robustness Toolbox* [Nicolae et al. 2019b]. Neste trabalho, focamos em dois principais métodos de defesa para ataques adversariais. São eles: Redução de Características do Modelo e Treinamento Adversarial..

5.1. Redução de Características do Modelo

Trabalhos como de [Bhagoji et al. 2017] e [Zhang et al. 2020] exploram o aumento na potência de ataques adversariais com base no número de *features* utilizadas pelos modelos

em questão. A alta dimensionalidade inerente aos dados, ou pela ausência de restrições na modificação, impacta a potência dos ataques adversariais, justificando a necessidade de métodos defensivos que gerenciem essa dimensionalidade ou limitem as *features*.

Essa relação é empiricamente explicada ao possuir mais *features* modificáveis, gerando oportunidades ao atacante para modificação de características chave que podem influenciar a saída do classificador. Portanto, o método adotado foi a seleção de *features* para redução da superfície de ataque. Este é um método defensivo simplificado capaz de mitigar ataques de evasão, em particular tratando-se de modelos de aprendizado profundo que operam com uma alta dimensionalidade de dados. De modo geral, o método de redução de características reduz as dimensões do modelo e o deixa menos suscetível a entradas que geram inferência errada.

5.2. Treinamento Adversarial

O treinamento adversarial é outro método defensivo amplamente utilizado para garantir a robustez do modelo contra ataques adversariais. O treinamento adversarial consiste na adoção de exemplos adversariais descobertos durante a fase de treinamento do modelo. Isso permite ampliar a base de dados do modelo garantindo que ele aprenda os padrões dos ataques, diminuindo o sucesso de ataques dessa magnitude contra o modelo em questão.

De forma geral, a ideia do treinamento adversarial é melhorar a robustez do classificador ao incorporar amostras adversariais no conjunto de treinamento. Quando múltiplos ataques são considerados, é possível combinar diferentes amostras adversariais para compor um conjunto de dados aumentado, sendo então utilizado no treinamento de um classificador mais robusto. Essa ideia de utilizar amostras adversariais provenientes de diferentes ataques também foi explorada por estudos anteriores [Tramèr et al. 2020].

6. Configuração do Ambiente e Implementação

Esta seção apresenta como o ambiente real de testes foi configurado. A Seção 6.1 apresenta o ambiente de testes Open RAN e a base de dados considerada. A Seção 6.2 detalha a implementação dos ataques de evasão e os métodos de defesa.

6.1. Ambiente Open RAN e Base de Dados da xApp IDS

Antes de implementar o ataque de evasão DDoS-ADE, é necessário ter uma xApp que atue como IDS no Near-RT RIC. Essa xApp IDS não é disponibilizada nativamente por nenhuma ferramenta ou *framework*, com isso foi utilizada uma xApp baseada no trabalho de [Silva et al. 2025]. Por isso, previamente ao ataque, foi preciso criar uma base de dados com tráfego benigno e DDoS por UEs. Após essa fase, os dados foram rotulados e um modelo de IA/ML foi treinado para detectar tráfego benigno ou DDoS. Vale destacar que existem várias bases de dados para treinamento de IDS para detecção de ataques DDoS, porém tais *datasets* não foram gerados em uma rede Open RAN e por isso não refletem com precisão um cenário Open RAN.

A base de dados criada neste trabalho possui 62.495 amostras, composta por tráfego benigno e por ataques DDoS do tipo SYN. O tráfego benigno foi coletado a partir de aplicações reais de alto consumo de banda, como *streaming* de vídeos em 4K e transmissões ao vivo. Por outro lado, os ataques DDoS simulam conexões SYN maliciosas, típicas de tentativas de exaustão de recursos do servidor. No *dataset* estão presentes 16 *features* de métricas O-RAN, que representam métricas relacionadas ao desempenho da rede, tais como: latência de indicação (IndLatency), taxa de sucesso de pacotes

(DRB.PacketSuccessRateUlgNBUu), volume transmitido na camada de controle de link de rádio (DRB.RlcSduTransmittedVolumeDL/UL), vazão por usuário (DRB.UEThpDl, DRB.UEThpUl), e métricas sobre o uso de blocos de recursos (RRU.PrbUsedDl e RRU.PrbAvailUl), o número de características disponíveis é limitada pela pilha de software da RAN. Quanto à distribuição de classes, 55.553 das amostras correspondem a tráfego benigno, enquanto 6.942 representam tráfego malicioso do tipo DDoS SYN.

A fase de coleta de dados envolvendo o tráfego das UEs foi feito utilizando a infraestrutura do *testbed* OpenRAN@Brasil, no site da Rede Nacional de Ensino e Pesquisa (RNP). Neste, foram utilizados dois servidores denominados O-RAN Cloud 2 e O-RAN Cloud 5. Em ambos os servidores foram instalados um kernel *realtime* recomendados para a execução das funções de RAN CU/DU e do núcleo 5G (5GC). Além disso, foram instanciados dois *clusters* Kubernetes para cada um dos servidores, sendo este requisito para a instalação dos *containers* que envolvem a RAN, 5GC e Near-RT RIC. As especificações detalhadas dos servidores estão descritas na Tabela 1.

Tabela 1. Especificações Servidores Open RAN.

Servidores	**O-RAN Cloud 2**	**O-RAN Cloud 5**
CPU Intel(R) Xeon(R)	Silver 4314 CPU @ 2.40GHz	Gold 6348 CPU @ 2.60GHz
Memória	128 GB	256G
Armazenamento	500 GB	500 GB
S.O	Ubuntu 24.04.2 LTS	Ubuntu 24.04.2 LTS
Kernel	6.8.1-1018-realtime	6.8.1-1017-realtime

Com relação a RU, foi utilizada a RPQN-7801E da Foxconn configurada para o modo de operação em banda n78 e utilizando os parâmetros mostrados na Tabela 2. À RU foi conectado um Raspberry PI 4 com um módulo Hat 5G SIM8262E-M2, no qual, neste cenário, possui o papel de UE. Estes equipamentos foram cedidos e pré-configurados com suporte da RNP e foram alocados em um ambiente *indoor* no *testbed* OpenRAN@Brasil.

Tabela 2. Modo de Operação O-RU

O-RU Foxconn RPQN-7801E	
Frequência de Operação	3.74 GHz
Largura de Banda	100 MHz
Nº PRBs	273
Modo de Transmissão	4T4R MIMO
Espaçamento entre Subportadora	30 KHz

No que diz respeito as pilhas de software, para a implantação da O-CU e O-DU, o projeto srsRAN [SRS 2025] da *Software Radio Systems* foi utilizado, suportando o modelo de serviço E2SM-KPM, crucial para o monitoramento do tráfego normal e malicioso na rede 5G. Para a implantação do núcleo da rede, foi utilizado o Open5GS, no qual é um projeto de código aberto que implementa as principais funções responsáveis pelo acesso, autenticação e troca de dados do UE com a rede 5G. A Figura 5 ilustra o ambiente montado para a realização da coleta de dados.

O processo de coleta de métricas envolveu uma xApp, escrita em Golang, que utiliza o SDK do OSC RIC para o envio de solicitação de subscrição para a O-CU e a O-DU. Após o envio e o aceite da subscrição por parte das funções da RAN, as métricas

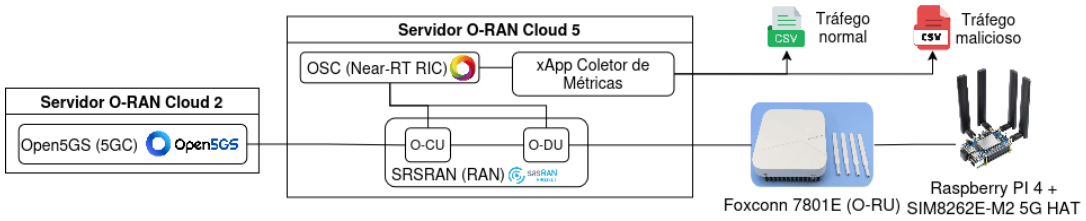

Figura 5. Arquitetura de Coleta no *testbed* OpenRAN@Brasil.

são enviadas por estas via interface E2 e, em seguida, recebidas pelo xApp por meio do RIC. O coletor de métricas, por sua vez, recebe os parâmetros de tráfego gerados no UE e armazena-os em dois *datasets* distintos que são dedicados aos tráfegos normais e maliciosos, respectivamente. Estes dados foram usados para a implementação dos ataques de evasão, bem como, dos modelos de AM com a defesa adversarial.

6.2. Implementação do Ataque de Evasão e os Métodos Defensivos

Na implementação do ataque adversarial DDoS-ADE, criamos duas versões, uma usando o FGSM e outra baseada no PGD. Ambas versões foram implementadas em Python 3.12.3, utilizando como base o projeto [Nicolae et al. 2019b] e o *framework* TensorFlow 2.19.0, permitindo a geração de exemplos adversariais diretamente sobre os dados de validação. O FGSM foi aplicado como uma única perturbação proporcional ao sinal do gradiente da função de perda, e somado a entrada original, controlada pelo parâmetro ε, que foi variado de 0.1 a 1.0. O PGD foi implementado como uma versão iterativa do FGSM, realizando 40 iterações com passo de atualização $\alpha = 0,01$. Os dois ataques foram aplicados nas amostras da classe de ataque DDoS.

Além disso, neste trabalho implementamos os seguintes métodos de defesa, os quais foram aplicados ao xApp IDS:

- **D.1** Esta defesa aplica o método de Redução de Características do Modelo. Portanto, o modelo de AM foi reduzido de forma que não prejudique significativamente métricas de desempenho como Acurácia, Precisão, Recall e F1-Score.
- **D.2** Esta defesa baseia-se no método de Treinamento Adversarial. Desta forma, com base nos exemplos adversariais que foram encontrados capazes de evadir a detecção DDoS, houve um retreinamento do modelo, incluindo na base de treino as amostras adversariais com o rótulo correto.
- **D.3** Esta defesa é a aplicação em conjunto das defesas D.1 e D.2.

7. Avaliação e Resultados Obtidos

Os resultados obtidos estão separados em duas subseções, a primeira parte é referente às métricas de desempenho dos modelos durante o processo de treinamento, presentes na Subseção 7.1. Por sua vez, os resultados do impacto dos ataques e de como as defesas implementadas se adaptaram nesse contexto estão presentes na Subseção 7.2.

7.1. Análise dos modelos

Os resultados dessa etapa se deram de forma iterativa com os testes da Subseção 7.2, possibilitando analisar o impacto dos ataques com diferentes tipos de abordagem para defesa. Para isso, primeiro foi realizado o treinamento do modelo LSTM (*Long Short-Term Memory*) sem nenhum tipo de defesa, o treinamento foi conduzido utilizando a validação

cruzada K-Fold com 5 divisões, a fim de garantir a representatividade das classes em todas as partições dos dados. Para a etapa de inferência, adotou-se o modelo LSTM devido à sua comprovada eficácia na análise de dados sequenciais em sistemas de detecção de intrusão, conforme demonstrado em trabalhos relacionados [Chang et al. 2024]. Sua arquitetura é capaz de capturar dependências de longo prazo e reconhecer padrões temporais, características essenciais para a identificação de comportamentos anômalos associados a ataques DDoS.

A construção do modelo LSTM foi composta por duas camadas LSTM com 64 e 32 unidades, em intercalação com camadas de *dropout* para reduzir o *overffiting*. Além disso, foi utilizada uma camada densa com ativação sigmoide para a classificação binária. O modelo foi treinado por 20 épocas com *batch* de 128, utilizando a função de perda *binary_crossentropy* e o otimizador Adam, foram utilizados pesos de classe com base na distribuição das amostras para mitigar o desequilíbrio. No que se refere ao balanceamento entre as classes de tráfego benigno e ataques SYN flood, utilizou-se a técnica StratifiedK-Fold durante a validação cruzada, garantindo que cada *fold* preservasse a proporção original das classes. Além disso, aplicaram-se as técnicas de SMOTE e *data augmentation* para ampliar a representação da classe minoritária (ataques), mitigando o viés em favor da classe majoritária.

A Tabela 3 detalha o desempenho médio de cada *fold*, onde o modelo apresenta um alto Recall ao que diz respeito a classe 1, referente ao ataque DDoS. Adotar esse método de treinamento é importante, por permitir selecionar qual o melhor *fold* e posteriormente treinar o modelo completo o utilizando como referência. Com isso, temos a primeira série de modelos sem defesa que serão avaliados contra os ataques de evasão.

Tabela 3. Resultados de Precision, Recall e F1-score dos modelos (média folds).

Classe	**Sem Defesa**			**Defesa 1**			**Defesa 2**			**Defesa 3**		
	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1
Benigna - 0 (%)	99,87	99,66	99,77	99,80	98,69	99,24	99,88	99,67	99,77	99,84	95,82	97,79
DDoS - 1 (%)	97,33	98,98	98,14	90,41	98,40	94,23	97,38	99,01	98,18	74,71	98,78	85,07
Acurácia (%)	99,58			98,66			99,60			96,15		
Média Macro (%)	98,60	99,32	98,96	95,10	98,55	96,73	98,63	99,34	98,98	87,28	97,30	91,43
Média Ponderada (%)	99,59	99,58	99,59	98,76	98,66	98,68	99,60	99,60	99,60	97,05	96,15	96,38

Para implementar a primeira defesa adversarial, redução de características, foi realizado um estudo dos impactos que cada característica, utilizada para inferência, tinha no modelo, com isso foi identificado que realizando a redução das 10 métricas utilizadas para as 3 principais (IndLatency, DRB.UEThpDl, DRB.UEThpUl) não impactou de forma tão significa as métricas de desempenho do modelo como é visto na Tabela 3, principalmente no Recall para a classe 1. Os outros passos de treinamento se mantiveram os mesmos, resultando na segunda série de modelos que seriam avaliados contra os ataques de evasão.

Para implementar o segundo método de defesa, treinamento adversarial, foram gerados instâncias adversariais com base nos métodos de ataque FGSM e PGD, cada uma explorando diferentes intensidades de perturbações controladas pela variação de ε entre 0.1 e 1.0. As amostras adversariais produzidas foram combinadas com o conjunto de dados original, formando um conjunto de dados expandido, com essas instâncias novas sendo rotuladas como ataques, com isso, o modelo LSTM foi treinado, utilizando os mesmos padrões mencionados anteriormente. Esse procedimento foi realizado para cada divisão do K-Fold, a média de cada *fold* está presente na Tabela 3, podemos perceber uma pequena variação do Recall em relação ao modelo original para a classe 1, porém métricas

como a de precisão apresentam uma perda maior.

Por fim, a terceira defesa implementada consistiu na união dos dois métodos anteriores, dessa forma foi realizado o treinamento adversarial em um modelo com métricas reduzidas seguindo os passos explicados anteriormente, podemos ver na Tabela 3 que semelhante à adoção dos outros métodos de defesa se observou uma redução nas métricas, porém nada que impacte de forma negativa o funcionamento do modelo.

Outra análise fundamental, dado o ambiente real, é o tempo de treinamento ou retreinamento do modelo. A Tabela 4 demonstra a variação de tempo de cada método de treinamento adotado para este trabalho, onde temos a distribuição de cada tempo de cada *fold*. O primeiro método de defesa não apresentou impacto significativo no tempo de treinamento do novo modelo, observa-se uma redução no tempo devido à redução das métricas utilizadas. Entretanto, o treinamento adversarial implementado nos métodos de defesa 2 e 3 tiveram um impacto significativo aumentando o tempo de treinamento.

Tabela 4. Tempos por Fold e Comparação de Defesas

Defesa	Fold 1 (min)	Fold 2 (min)	Fold 3 (min)	Fold 4 (min)	Fold 5 (min)
Sem Defesa	2,0657	1,9830	2,0046	2,0448	2,0664
Defesa 1	1,9024	1,8340	1,9159	1,8991	1,8470
Defesa 2	87,8533	87,3567	88,5024	88,0710	90,3044
Defesa 3	85,9960	86,0748	87,5533	86,6298	87,4386

Porém, em Open RAN embora o ciclo de controle de uma aplicação que atua acoplada ao Near-RT RIC seja no máximo 1 segundo, o modelo de aprendizado de máquina não precisa obedecer esse ciclo, ao ser possível retreinar um modelo fora desse ciclo, adotando métodos que custem mais tempo. Além disso, durante o processo de treinamento adversarial foram gerados exemplos para os 2 ataques e suas variações de ε, a implementação gradual de exemplos para novos modelos pode ser uma abordagem válida.

7.2. Análise Impacto dos Ataques

A variação do parâmetro ε dita o grau de perturbação que as amostras serão submetidas. Valores baixos de ε podem gerar perturbações praticamente imperceptíveis, porém com menor eficácia adversarial, as grandes variações aumentam o risco de detecção, mas também aumentam a taxa de sucesso da evasão. Desse modo, o ataque foi testado com vários valores para ε.

A Figura 6 apresenta os resultados dos ataques FGSM e PGD em cada *fold* sem defesa, se percebe a variação de ε a partir de certo ponto inviabilizaria o funcionamento do modelo para detecção de ataques, foram utilizadas variações maiores de $\varepsilon =< 5$, porém, a partir do limiar $\varepsilon = 1$ a degradação do desempenho do modelo apresentou uma baixa variação.

A métrica adotada para avaliar o impacto do ataque no modelo de AM é o Recall da classe 1, pois ela permite medir a capacidade de um modelo de identificar corretamente os ataques, quanto menor o Recall, significa que muitos ataques estão passando pelo modelo como tráfego benigno. Desse modo, foi retirada a média do Recall referente aos 5 Folds. O ataque PGD apresentou um maior impacto no modelo sem necessitar de grande variações no valor de ε, apresentando um comportamento condizente com sua maior complexidade, porém os ataques mais simples como FGSM apresentam uma grande eficácia.

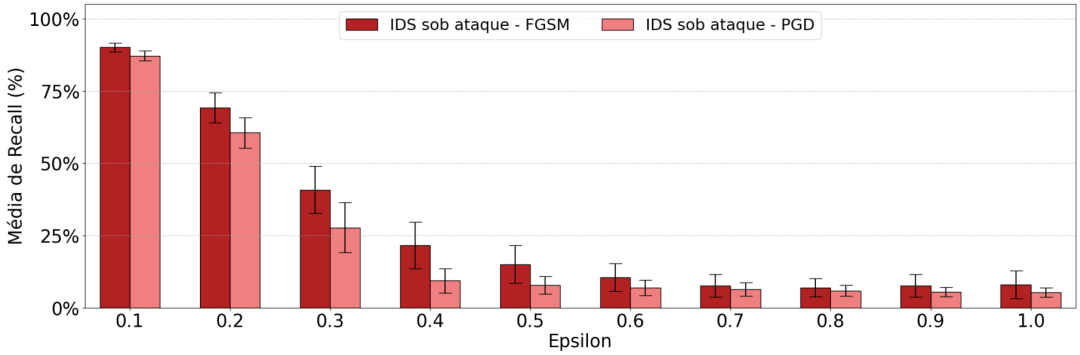

Figura 6. Impacto dos ataques realizados no Recall do modelo LSTM sem defesa.

Outro ponto, representado na Figura 6, é o alto valor do desvio padrão em relação aos valores do Recall para cada *fold*, portanto, dependendo do conjunto de dados para o treino em cada *fold*, o modelo pode apresentar uma maior resistência a essas alterações.

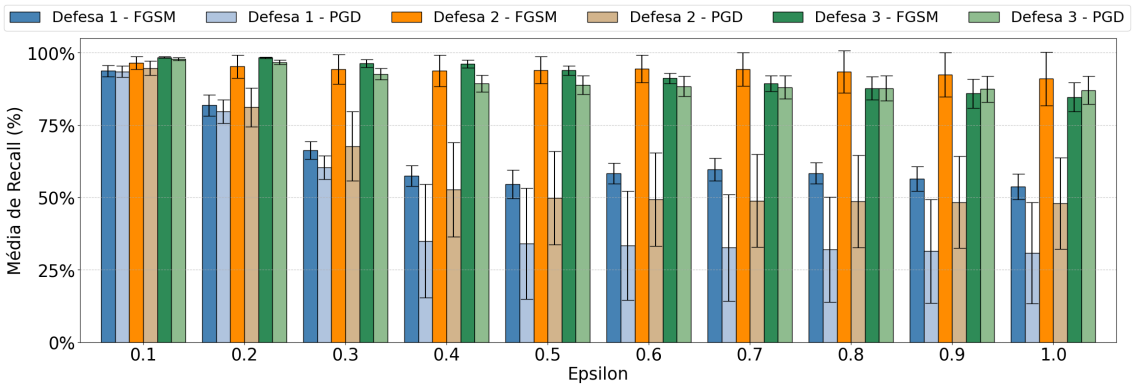

Figura 7. Impacto dos ataques realizados no Recall do modelo LSTM com defesa.

A Figura 7 apresenta os resultados obtidos em relação aos métodos de defesa adotados, o método de defesa 1 trouxe um ganho significativo na confiabilidade do modelo contra os ataques, principalmente contra os ataques FGSM, porém em relação ao ataque mais robusto como o PGD obteve uma baixa taxa de sucesso. Desse modo, é possível afirmar que durante o processo de treinamento somente as características essenciais devem ser consideradas, realizando um balanço entre a perda de desempenho do modelo em algumas métricas e a sua segurança.

Os resultados da defesa 2 demonstram a eficiência do treinamento adversarial para ambos os casos. Desse modo, evidenciando sua maior robustez frente a perturbações, mesmo aquelas geradas por métodos de ataques mais sofisticados como o PGD. A média de Recalls obtidos foi superior a defesa 1, um resultado lógico dada a simplicidade do primeiro método de defesa. Por fim, a defesa 3 apresentou os melhores resultados e menor desvio padrão, isso evidencia a eficiência da adoção de múltiplos métodos de defesa para aumentar a robustez dos modelos frente a diferentes ataques adversariais, a combinação de estratégias de proteção é mais eficaz do que a aplicação isolada de técnicas individuais.

8. Conclusão e Trabalhos Futuros

Em suma, com os resultados obtidos foi possível evidenciar os riscos do ataque adversariais de evasão proposto, denominado de *Distributed DoS Adversarial Detection Evasion*

(DDoS-ADE), em um cenário crítico de Open RAN e uma xApp IDS especializada em ataques DDoS. As análises demonstram que ataques simples, como o FGSM, já são capazes de comprometer significativamente o desempenho de modelos de AM, enquanto ataques mais robustos, como PGD, aumentam ainda mais essa vulnerabilidade.

A implementação de três defesas permitiu avaliar a eficácia de métodos de diferentes complexidades, evidenciando também que abordagens simples, como a seleção de características mais relevantes, já trazem ganhos significativos na robustez do modelo. Por sua vez, métodos mais elaborados, como o treinamento adversarial, apresentam maior resistência frente a perturbações e a combinação de diferentes defesas permite um ganho ainda maior na confiabilidade do modelo frente a esses ataques.

Assim, concluindo-se que adoção de múltiplas estratégias de defesa é crucial para garantir níveis mais elevados de confiabilidade e resiliência para sistemas baseados em AM dentro do cenário Open RAN. Além disso, mesmo que os tempos de treinamento de técnicas de defesa mais robustas, como o treinamento adversarial, tenham se mostrado bem maiores, ainda permanecem viáveis em Open RAN, ao ser possível integrar o retreino fora do ciclo crítico de operação empregado pelo Near-RT RIC.

Em relação aos trabalhos futuros, algumas direções importantes foram identificadas. Primeiro, o retreinamento contínuo e de forma incremental será investigado, pois embora o treinamento adversarial tenha apresentado ótimas métricas, ele não torna o modelo robusto contra ataques que não estavam previstos anteriormente em seu treinamento, com isso se tem a previsão de adotar o *framework* de retreino da arquitetura Open RAN para adicionar novas iterações de ataques ao decorrer do tempo para cobrir uma gama de ataques maior. Outro trabalho que será realizado está na validação desse cenário com múltiplas UEs, e verificar o impacto no tempo de resposta do IDS que cada método de defesa tem em diferentes cargas de ataques DDoS e ataques de evasão.

Por fim, os resultados apresentados no artigo estão presentes no repositório do projeto da equipe [Dias 2025], contendo o conjunto de dados que seguirá sendo atualizado e os *scripts* de treinamento e ataques realizados, possibilitando uma contribuição com a comunidade e fomentando esta área de pesquisa.

9. Agradecimentos

Este trabalho foi parcialmente financiado pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) projeto 444978/2024-0, pela Fundação Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), e pela Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP) projeto 2023/00811-0, projeto 2023/00673-7, projeto 2021/00199-8 (CPE SMARTNESS), projeto 2020/04031-1, e projeto 2018/23097-3. Ao OpenRAN@Brasil em parceria com a Rede Nacional de Ensino e Pesquisa (RNP) e Ministério de Ciência e Tecnologia (MCTI) projeto 01245.014203/2021-14.

Referências

[Alliance 2021] Alliance, O.-R. (2021). O-ran ai/ml workflow description and requirements 1.03. Relatório Técnico O-RAN.WG2.AIML-v01.03, O-RAN Alliance. Acessado em 20 de janeiro de 2025.

[Alliance 2025a] Alliance, O.-R. (2025a). O-ran security threat modeling and risk assessment 5.0. Relatório Técnico O-RAN.WG11.TR.Threat-Modeling.O-R004-v05.00, O-RAN Alliance. Acessado em 10 de Fevereiro de 2025.

[Alliance 2025b] Alliance, O.-R. (2025b). O-ran study on security for artificial intelligence and machine learning (ai/ml) in o-ran 3.0. Relatório Técnico O-RAN.WG11.TR.AIML-Security-Analysis.0-R004-v03.00, O-RAN Alliance. Acessado em 15 de Fevereiro de 2025.

[Amachaghi et al. 2024] Amachaghi, E. N., Shojafar, M., Foh, C. H., and Moessner, K. (2024). A survey for intrusion detection systems in open ran. *IEEE Access*, 12:88146–88173.

[Ayub et al. 2020] Ayub, M. A., Johnson, W. A., Talbert, D. A., and Siraj, A. (2020). Model evasion attack on intrusion detection systems using adversarial machine learning. In *2020 54th Annual Conference on Information Sciences and Systems (CISS)*, pages 1–6.

[Bhagoji et al. 2017] Bhagoji, A. N., Cullina, D., Sitawarin, C., and Mittal, P. (2017). Enhancing robustness of machine learning systems via data transformations.

[Chang et al. 2024] Chang, J.-E., Chiu, Y.-C., Ma, Y.-W., Li, Z.-X., and Shao, C.-L. (2024). Packet continuity ddos attack detection for open fronthaul in oran system. In *NOMS 2024-2024 IEEE Network Operations and Management Symposium*, pages 1–5.

[Costa et al. 2024] Costa, J. C., Roxo, T., Proença, H., and Inácio, P. R. M. (2024). How deep learning sees the world: A survey on adversarial attacks defenses. *IEEE Access*, 12:61113–61136.

[Dias 2025] Dias, V. (2025). Usap-5g - sid-xapp branch. `https://github.com/muriloAvlis/USAP-5G/tree/SID-xApp`.

[Ergu et al. 2025] Ergu, Y. A., Nguyen, V.-L., Hwang, R.-H., Lin, Y.-D., Cho, C.-Y., Yang, H.-K., Shin, H., and Duong, T. Q. (2025). Efficient adversarial attacks against drl-based resource allocation in intelligent o-ran for v2x. *IEEE Transactions on Vehicular Technology*, 74(1):1674–1686.

[Marinova and Leon-Garcia 2024] Marinova, S. and Leon-Garcia, A. (2024). Intelligent o-ran beyond 5g: Architecture, use cases, challenges, and opportunities. *IEEE Access*, 12:27088–27114.

[Nicolae et al. 2019a] Nicolae, M.-I., Sinn, M., Tran, M. N., Buesser, B., Rawat, A., Wistuba, M., Zantedeschi, V., Baracaldo, N., Chen, B., Ludwig, H., Molloy, I. M., and Edwards, B. (2019a). Adversarial robustness toolbox v1.0.0.

[Nicolae et al. 2019b] Nicolae, M.-I., Sinn, M., Tran, M. N., Buesser, B., Rawat, A., Wistuba, M., Zantedeschi, V., Baracaldo, N., Chen, B., Ludwig, H., Molloy, I. M., and Edwards, B. (2019b). Adversarial robustness toolbox v1.0.0.

[O-RAN Alliance 2018] O-RAN Alliance (2018). O-ran whitepaper - building the next generation ran. Whitepaper, O-RAN Alliance.

[Polese et al. 2023] Polese, M., Bonati, L., D'Oro, S., Basagni, S., and Melodia, T. (2023). Understanding o-ran: Architecture, interfaces, algorithms, security, and research challenges. *IEEE Communications Surveys Tutorials*, 25(2):1376–1411.

[Rimedo Labs 2023] Rimedo Labs (2023). Rimedo labs. `https://rimedolabs.com/blog/ml-framework-in-o-ran/`.

[Sapavath et al. 2023] Sapavath, N. N., Kim, B., Chowdhury, K., and Shah, V. K. (2023). Experimental study of adversarial attacks on ml-based xapps in o-ran. In *GLOBECOM 2023 - 2023 IEEE Global Communications Conference*, pages 6352–6357.

[Sharafaldin et al. 2018] Sharafaldin, I., Habibi Lashkari, A., and Ghorbani, A. (2018). Toward generating a new intrusion detection dataset and intrusion traffic characterization. pages 108–116.

[Sharafaldin et al. 2019] Sharafaldin, I., Lashkari, A. H., Hakak, S., and Ghorbani, A. A. (2019). Developing realistic distributed denial of service (ddos) attack dataset and taxonomy. In *2019 International Carnahan Conference on Security Technology (ICCST)*, pages 1–8.

[Silva et al. 2025] Silva, M., Oliveira, L., Dias, V., Gomes, M., Farias, F., Riker, A., and Abelém, A. (2025). Automatizando a alocação de usuários em slices 5g em arquiteturas open ran. In *Anais do XXX Workshop de Gerência e Operação de Redes e Serviços*, pages 127–140, Porto Alegre, RS, Brasil. SBC.

[SRS 2025] SRS (2025). Open-source and ORAN-native 5G CU/DU with a complete stack from I/Q to IP from SRS.

[Szegedy et al. 2014] Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. (2014). Intriguing properties of neural networks.

[Tramèr et al. 2020] Tramèr, F., Kurakin, A., Papernot, N., Goodfellow, I., Boneh, D., and McDaniel, P. (2020). Ensemble adversarial training: Attacks and defenses.

[Xavier et al. 2023] Xavier, B. M., Dzaferagic, M., Collins, D., Comarela, G., Martinello, M., and Ruffini, M. (2023). Machine learning-based early attack detection using open ran intelligent controller. In *ICC 2023 - IEEE International Conference on Communications*, pages 1856–1861.

[Zhang et al. 2020] Zhang, S., Xie, X., and Xu, Y. (2020). A brute-force black-box method to attack machine learning-based systems in cybersecurity. *IEEE Access*, PP:1–1.