Provimento de Privacidade na Detecção de Anomalias na O-RAN*

Thiago Tokarski[1], **Eduardo Alchieri**[1], **Priscila Solis Barreto**[1]

[1] Departamento de Ciência da Computação, Universidade de Brasília (UnB)

thiago.tokarski@aluno.unb.br, {alchieri,pris}@unb.br

Abstract. *The Open Radio Access Network (O-RAN) architecture promises to transform wireless communication by enabling intelligent network management through third-party applications, known as xApps and rApps. These applications access sensitive network and user data, creating privacy and security risks. At the same time, anomaly detection is essential to ensure reliable performance and security in O-RAN. In this context, this paper proposes a privacy-preserving anomaly detection solution for O-RAN by using Fully Homomorphic Encryption (FHE) to obfuscate sensitive data before it is accessed by third-party xApps. Experimental results show that although FHE introduces computational overhead, the inference latency remains suitable for the O-RAN control loop. Furthermore, the model's accuracy is not affected, demonstrating that FHE is a viable alternative for building secure and trustworthy intelligence in O-RAN.*

Resumo. *A arquitetura Open Radio Access Network (O-RAN) promete transformar a comunicação sem fio ao permitir gestão inteligente da rede por meio de aplicativos de terceiros, conhecidos como xApps e rApps. Contudo, estes aplicativos acessam dados sensíveis da rede e dos usuários, criando riscos de privacidade e segurança. Ao mesmo tempo, a detecção de anomalias é essencial para garantir desempenho confiável e segurança da O-RAN. Nesse contexto, este artigo propõe uma solução de detecção de anomalias com preservação de privacidade para a O-RAN, através da utilização de Fully Homomorphic Encryption (FHE) para ocultar dados sensíveis antes de serem acessados por xApps de terceiros. Os resultados experimentais mostram que, embora a FHE cause sobrecarga computacional, a latência de inferência continua adequada para o ciclo de controle do O-RAN. Além disso, a acurácia do modelo não é afetada, demonstrando que a FHE é uma alternativa viável para construir inteligência segura e confiável na O-RAN.*

1. Introdução

A detecção de anomalias na O-RAN é essencial para manter a confiabilidade, segurança e eficiência dos sistemas modernos. As arquiteturas O-RAN são compostas por componentes desagregados e interoperáveis de múltiplos fornecedores, oferecendo maior flexibilidade mas também aumentando a complexidade e o potencial de vulnerabilidades. Implementar mecanismos eficazes de detecção de anomalias permite que operadores identifiquem padrões incomuns no comportamento da rede, como degradação de desempenho, erros de configuração ou atividades maliciosas. A proatividade não só melhora a

*Este trabalho é apoiado pela Fundação de Apoio à Pesquisa do Distrito Federal (FAP-DF) através dos projetos PLEDESIR e PI2RAFut e pela Rede Nacional de Ensino e Pesquisa (RNP) através do projeto OpenRan@Brasil.

qualidade do serviço como também permite a automação e otimização da rede, fatores fundamentais para o sucesso das RANs (*Radio Access Networks*) inteligentes.

O modelo desagregado proposto pela arquitetura O-RAN utiliza microsserviços conhecidos como xApps e rApps, que podem ser de terceiros, utilizados para automatizar e otimizar a gestão das redes O-RAN, possibilitando controle e tomada de decisão inteligentes em tempo real. Embora essa abordagem tenha como objetivo estimular a inovação, também apresenta desafios fundamentais de privacidade que há muito tempo afetam a indústria de computação em nuvem: como processar dados sensíveis por aplicações de terceiros não confiáveis. Nesse sentido, a crescente popularidade dos serviços baseados em nuvem tem acelerado o desenvolvimento de soluções criptográficas avançadas capazes de ajudar a resolver esse problema.

Um trabalho recente [Tsourdinis et al. 2024] demonstrou a viabilidade de um xApp baseado em IA (Inteligência Artificial) para detecção de intrusões em redes O-RAN. No entanto, esta abordagem processa dados sensíveis da rede em texto claro, levantando preocupações significativas de privacidade. De fato, a análise de ameaças de segurança da O-RAN Alliance [O-RAN ALLIANCE 2022] identifica explicitamente o acesso não regulamentado a dados confidenciais de usuários e da rede por xApps como uma ameaça crítica à arquitetura [Polese et al. 2023]. Estes xApps/rApps baseados em IA, que estão no centro dessa ameaça, normalmente dependem de modelos de aprendizado de máquina para classificar o tráfego de rede. Modelos baseados em árvores são uma escolha popular e poderosa para esse fim, especialmente para dados tabulares [Grinsztajn et al. 2022], comumente encontrados em métricas de desempenho da rede. Trabalhos recentes sobre segurança em O-RAN identificaram o modelo *Random Forest* (RF) como um classificador ideal para detectar ataques, alcançando alta acurácia em *testbeds* realistas [Xavier et al. 2023].

O ponto central deste desafio está relacionado com as limitações computacionais inerentes aos *frameworks* criptográficos convencionais. Embora mecanismos tradicionais de criptografia sejam indispensáveis para garantir a confidencialidade dos dados, eles são inadequados para cenários que exigem processamento nos dados. De fato, soluções baseadas nestes esquemas precisam descriptografar as informações antes da computação e existe um *trade-off* intrínseco entre habilitar inteligência avançada de rede e preservar garantias de privacidade. No contexto do O-RAN, isso significaria que um xApp/rApp de terceiros precisaria ter acesso a dados sensíveis. Felizmente, a *Fully Homomorphic Encryption* (FHE) oferece uma solução para esse dilema, pois é um método que permite executar computações diretamente sobre dados criptografados [Acar et al. 2018].

Nesse contexto, este artigo propõe uma solução para detecção de anomalias com preservação de privacidade na O-RAN, composta por um conjunto de xApps e rApps que utilizam FHE para ofuscar dados sensíveis antes que aplicativos de terceiros os acessem. A solução proposta é baseada nos conceitos apresentados em [Tsourdinis et al. 2024] e demonstra como a mesma lógica de detecção de anomalias pode ser executada sobre dados criptografados. A abordagem proposta garante que informações sensíveis de usuários e da rede permaneçam confidenciais e oferece um caminho para implantações O-RAN mais seguras e confiáveis. A proposta de um sistema baseado em FHE é avaliada e são analisados os *trade-offs* entre privacidade e sobrecarga computacional.

O restante deste artigo está organizado da seguinte forma. A Seção 2 apresenta

os fundamentos teóricos. A Seção 3 revisa os trabalhos relacionados. A Seção 4 apresenta a proposta de detecção de anomalias com preservação de privacidade em O-RAN. A Seção 5 descreve a avaliação experimental. Por fim, a Seção 6 conclui o artigo.

2. Fundamentos Teóricos

Esta seção apresenta os conceitos teóricos necessários para compreender a proposta da solução para detecção de anomalias com preservação de privacidade na O-RAN.

2.1. Random Forest e *Full Homomorphic Encryption*

Random Forest (RF) é um método de aprendizado de máquina que constrói várias árvores de decisão durante a fase de treinamento, com o objetivo de reduzir a variância e apresentar um melhor desempenho preditivo [Louppe 2014]. Cada nova entrada é processada por todas as árvores e a saída é a média de todas as predições individuais. Em uma tarefa de classificação, a classe final é escolhida por votação majoritária entre todas as árvores.

O comportamento do RF é determinado por diversos hiperparâmetros, sendo os mais relevantes considerados neste trabalho o número de árvores *(n_estimators)* e a profundidade máxima de cada árvore *(max_depth)*. O número de estimadores impacta diretamente a variância do modelo, pois aumentar o número de árvores ajuda a reduzir o componente de variância do erro de generalização, o que, por sua vez, melhora a acurácia. O parâmetro *max_depth* atua como um critério de parada para controlar a complexidade das árvores de decisão individuais. Ao limitar a profundidade, esse hiperparâmetro evita que as árvores se tornem excessivamente complexas e sofram *overfitting*.

Homomorphic Encryption (HE) é uma forma especializada de criptografia que permite a realização de computações diretamente sobre dados criptografados, sem a necessidade de descriptografá-los. *Full Homomorphic Encryption* (FHE) suporta um número ilimitado de diferentes operações realizadas um número ilimitado de vezes [Acar et al. 2018]. No domínio da FHE, o método utilizado para lidar com as funções não lineares das árvores de decisão é um fator diferenciador fundamental. Muitas abordagens se baseiam em esquemas de FHE, como Cheon-Kim-Kim-Song (CKKS) [Cheon et al. 2016] ou Brakerski/Fan-Vercauteren (BFV) [Fan and Vercauteren 2012], que normalmente exigem que a lógica da árvore seja aproximada por uma função polinomial. No entanto, este processamento pode alterar o comportamento do modelo e introduzir imprecisões. Uma alternativa atual evita essa aproximação ao utilizar o esquema TFHE (*Tree Full Homomorphic Encryption*) para realizar inferência exata e de alta fidelidade em modelos baseados em árvores [Frery et al. 2023]. Essa proposta foi implementada na biblioteca open-source "Concrete-ML" [Zama 2022].

A fase inicial para utilização deste esquema concentra-se na criação e preparação de um modelo de aprendizado de máquina para execução com FHE. A fase de implantação operacionaliza o modelo FHE, normalmente dentro de uma arquitetura cliente-servidor, permitindo que o servidor realize computações sobre dados criptografados fornecidos pelos usuários. Após isso, quatro operações centrais constituem o fluxo de trabalho:

- Geração de chaves (K): O processo é iniciado pelo cliente, que é responsável pela geração das chaves. Essa etapa cria as chaves criptográficas: uma chave privada, que é mantida em sigilo pelo cliente para criptografar as entradas e descriptografar as saídas; e uma chave pública de avaliação, que é compartilhada com o servidor para permitir computações sobre dados criptografados.

- Preparação de dados pelo cliente (E): Antes de serem enviados ao servidor para inferência, os dados de entrada do cliente passam por uma sequência de preparação em três etapas. Primeiro, os dados são quantizados para um formato inteiro compatível com o modelo FHE. Em seguida, essa representação inteira é criptografada utilizando a chave privada. Por fim, o texto cifrado resultante é serializado em um formato de dados adequado para transmissão.
- Predição criptografada (P): O servidor recebe o texto cifrado do cliente e executa a tarefa central de predição ao executar seu modelo sobre os dados criptografados. Um princípio fundamental dessa operação é que o servidor gera um resultado criptografado sem ter acesso às informações em texto claro.
- Processamento do resultado (D): Após a conclusão da predição, o resultado criptografado é enviado de volta ao cliente. O cliente então executa o processo inverso da sequência de preparação para obter a saída final. Os dados recebidos são descriptografados utilizando a chave privada para obtenção do resultado da predição.

2.2. Arquitetura Open RAN

A Figura 1 ilustra a arquitetura Open RAN (ou O-RAN) com uma visão esquemática resumida dos seus principais componentes e das interfaces de comunicação. A O-RAN representa uma mudança de paradigma em relação aos componentes RAN tradicionais e monolíticos. A visão O-RAN promove um novo modelo para o projeto e a implantação de redes baseado em componentes desagregados, virtualizados e orientados por software [Polese et al. 2023].

Esse novo paradigma é construído sobre quatro princípios fundamentais: desagregação, virtualização, interfaces abertas e controle inteligente orientado por dados. Ao desagregar a estação rádio-base em diferentes unidades funcionais (O-CU (*Central Unit*), O-DU (*Distributed Unit*) e O-RU (*Radio Unit*), o O-RAN permite que componentes de diferentes fornecedores sejam interoperáveis por meio de interfaces abertas e padronizadas. A O-RAN Alliance trabalha na padronização dessa arquitetura e fornece as especificações para implementar os princípios do Open RAN sobre redes móveis celulares. Essa abordagem aberta e definida por software possibilita a integração de controle em malha fechada orientado por dados, que constitui a base para a automação inteligente de redes [Polese et al. 2023].

Uma inovação fundamental da arquitetura O-RAN é a introdução dos RAN *Intelligent Controllers* (RICs), que são componentes programáveis projetados para executar rotinas de otimização e viabilizar o controle da RAN em malha fechada e orientado por dados. Os RICs obtêm uma visão abstrata e centralizada da rede ao processar fluxos de dados que transportam *Key Performance Measurements* (KPMs) provenientes da infraestrutura RAN. Esses dados são utilizados por algoritmos de IA e aprendizado de máquina para aplicar políticas de controle aos nós da RAN [Polese et al. 2023]. A arquitetura O-RAN especifica dois RICs distintos, que operam em diferentes escalas de tempo:

***Near-Real-Time* (Near-RT) RIC.** Este controlador é implantado na borda da rede e executa laços de controle com latência entre 10 milissegundos e 1 segundo, interagindo diretamente com os nós da RAN (O-CUs e O-DUs) por meio da interface E2. Este controlador hospeda aplicações de terceiros, conhecidas como xApps, para realizar o gerenciamento de recursos de rádio em tempo quase real [Polese et al. 2023].

***Non-Real-Time* (Non-RT) RIC.** Este controlador faz parte do marco de referência do

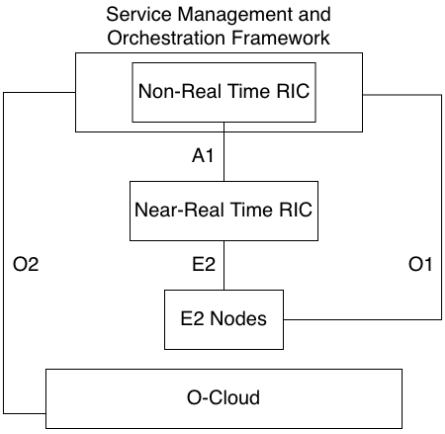

Figura 1. Visão resumida da O-RAN.

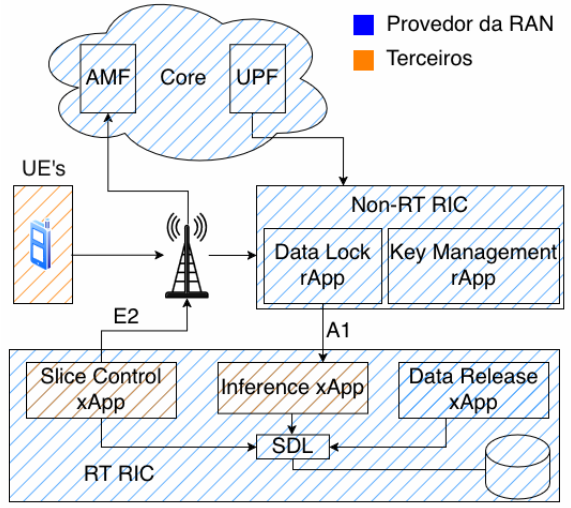

Figura 2. Solução proposta

Service Management and Orchestration (SMO) e opera em uma escala de tempo superior a 1 segundo. Suas principais funções incluem a definição de políticas, o fornecimento de dados e o gerenciamento dos modelos de IA/ML (na forma de rApps) que são implantados no Near-RT RIC [Polese et al. 2023]. Consequentemente, o Non-RT RIC atua como uma plataforma de software para rApps, permitindo a otimização de alto nível da RAN em termos de gerenciamento e controle [Dryjanski et al. 2021, Lacava et al. 2023].

3. Trabalhos Relacionados

Recentemente foi demonstrada uma implantação real de O-RAN para detecção de anomalias utilizando IA/ML [Tsourdinis et al. 2024]. Outros trabalhos propuseram xApps para identificar e mitigar ataques DDoS que podem comprometer fatias da rede O-RAN [Dias et al. 2024, Khan et al. 2022]. Essas abordagens também se baseiam em modelos tradicionais de aprendizado de máquina, sem preservação de privacidade, para detectar padrões maliciosos. Ambos os trabalhos evidenciam a tendência de incorporar inteligência à RAN, mas não abordam as implicações de privacidade do processamento de dados sensíveis da rede. De maneira semelhante, Xavier et al. [Xavier et al. 2023] desenvolveram um xApp no Near-RT RIC para realizar a detecção precoce de ataques de DoS utilizando dados das camadas física e MAC da RAN. Ao avaliar múltiplos algoritmos de aprendizado de máquina, eles identificaram o Random Forest como o modelo ideal, alcançando 95% de acurácia em um *testbed* realista em apenas 2.86 ms.

A necessidade de segurança e privacidade em O-RAN vai além da prevenção de acesso não autorizado aos dados, estendendo-se à defesa contra ameaças ativas originadas no próprio RIC. A arquitetura aberta, embora estimule a inovação, também introduz vetores de ataque nos quais xApps maliciosas ou comprometidas podem manipular o sistema. Uma análise sistêmica demonstrou essa vulnerabilidade ao implantar um xApp malicioso projetado para envenenar KPMs armazenados no banco de dados compartilhado do RIC [Chiejina et al. 2024]. O trabalho ressalta a necessidade de proteções criptográficas que não apenas preservam a privacidade dos dados em relação ao próprio xApp, mas também asseguram a integridade da computação contra entradas manipuladas.

A abordagem de ZT-RIC [Lin et al. 2025] utiliza Criptografia Funcional de Pro-

duto Interno (*Inner Product Functional Encryption*–IPFE) para desenvolver um xApp de detecção de *jammers*. A IPFE é especializada no cálculo de produtos internos sobre dados criptografados, o que pode ser altamente eficiente para certos modelos de aprendizado de máquina, como as camadas iniciais de redes neurais. O ZT-RIC alcançou uma latência de 0.527 segundos, atendendo aos rigorosos requisitos temporais do Near-RT RIC. Isso contrasta com a natureza mais geral da FHE, que suporta uma gama mais ampla de computações (como o percurso completo de uma árvore de decisão), porém geralmente com um custo computacional mais elevado. Esse cenário evidencia um *trade-off* fundamental entre a especificidade e eficiência da IPFE e a generalidade e flexibilidade da FHE para o desenvolvimento de xApps com preservação de privacidade.

Pela discussão anterior, os xApps são capazes de identificar anomalias e acionar ações subsequentes na rede. No entanto, um elemento comum nestas implementações é a dependência do processamento de dados sensíveis da rede em texto claro, que ignora riscos significativos de privacidade inerentes à arquitetura O-RAN. Assim, embora a integração funcional entre detecção de ameaças e mitigação automatizada tenha se mostrado viável, o aspecto crucial da privacidade dos dados permanece um grande desafio. Este trabalho aborda essa lacuna, ao propor uma solução baseada em um conjunto de xApps/rApps que introduz uma nova camada de preservação de privacidade por meio de FHE, para garantir que a detecção de anomalias possa ser realizada sem expor dados confidenciais da rede ou dos usuários.

4. Provendo Privacidade na Detecção de Anomalias na O-RAN

Esta seção apresenta a proposta de detecção de anomalias com preservação de privacidade em O-RAN. A proposta separa as responsabilidades de manipulação de dados, computação segura e aplicação de políticas entre múltiplos xApps/rApps, em conformidade com a abordagem de microsserviços utilizada na O-RAN. A Figura 2 mostra a arquitetura geral da solução proposta. O fluxo de trabalho foi projetado para garantir que os dados sensíveis sejam criptografados antes de serem expostos a qualquer componente de terceiros, processados integralmente em sua forma criptografada e descriptografados apenas por uma xApp confiável após a conclusão da análise. A separação de responsabilidades garante a confidencialidade dos dados ao longo de todo o pipeline de processamento. A seguir, é apresentado um resumo do procedimento passo a passo:

1. Coleta de Dados: Um servidor de captura de pacotes, co-localizado com a *User Plane Function* (UPF), monitora e agrega dados de fluxos de rede.
2. Criptografia dos Dados: Esses dados são encaminhados para a rApp Data Lock, que criptografa os dados sensíveis e envia o texto cifrado resultante para a xApp de Inferência por meio da interface A1 *Enrichment Information* (AIEI).
3. Inferência Homomórfica: O xApp de inferência recebe os dados criptografados, realiza a classificação baseada em FHE diretamente sobre o texto cifrado e grava a predição criptografada resultante no banco de dados SDL (*Shared Data Layer*).
4. Descriptografia da Classificação: O xApp confiável Data Release consulta o SDL e descriptografa a predição criptografada e atualiza o registro no SDL com o resultado final em texto claro.
5. Controle de *Slices*: Este xApp consulta o SDL em busca dos resultados em texto claro, agrega essas classificações para identificar ameaças e, em seguida, envia mensagens de controle à RAN por meio da interface E2 para mitigá-las.

A arquitetura explora componentes-chave das plataformas Non-RT RIC e Near-RT RIC e propõe um novo padrão arquitetural para a implantação de xApps que utilizam dados sensíveis, tais como informações de usuários finais ou dados da RAN que o operador não deseja compartilhar com xApps de terceiros.

4.1. Algoritmos dos xApps/rApps

A seguir, é descrita a função de cada componente. O Non-RT RIC hospeda dois rApps confiáveis (por exemplo, que foram desenvolvidos pelos operadores de rede):

rApp de Gerenciamento de Chaves: Este rApp é responsável por gerar e gerenciar de forma segura as chaves criptográficas da FHE. A chave pública é disponibilizada para qualquer xApp que necessite realizar inferência baseado em FHE (por exemplo, o xApp de Inferência). Além disso, este rApp transmite de forma segura a chave privada para os aplicativos confiáveis Data Lock e Data Release, que são os únicos componentes autorizados a criptografar dados da rede e a descriptografar os resultados, respectivamente.

rApp Data Lock: Este rApp (Algoritmo 1) atua como um gateway confiável para dados em texto claro. Sua responsabilidade é receber dados sensíveis da rede (como os KPMs), criptografá-los utilizando a chave privada e enviar o texto cifrado resultante diretamente para a xApp de Inferência por meio da interface A1.

Algoritmo 1 Data Lock rApp

Init
    Load data preprocessor
    Load FHE client model
EndInit
MainLoop
    Read $packet\_data$ from external enrichment server
    if $packet\_data$ **then**
        $model\_input \leftarrow preprocessor(packet\_data)$
        $encrypted\_input \leftarrow E(private\_key, model\_input)$
        Send $encrypted\_input$ to inference xApp through A1EI (Enrichment Information) interface
    end if
EndMainLoop

Já no Near-RT RIC, a arquitetura proposta utiliza três xApps distintos para gerenciar de forma segura o fluxo de inferência, separando computação da inferência, descriptografia do resultado e execução do controle da rede:

xApp de Inferência: Trata-se de uma aplicação de terceiros projetada para realizar detecção de anomalias. Conforme detalhado no Algoritmo 2, este xApp recebe a chave pública de avaliação a partir do rApp de Gerenciamento de Chaves e os dados criptografados do rApp Data Lock. Sua função central é executar o modelo de aprendizado de máquina habilitado para FHE, e fazer a predição diretamente sobre o texto cifrado. Como possui apenas a chave pública, este xApp não tem acesso ao resultado em texto claro. Após a computação, a predição criptografada é gravada no SDL.

xApp Data Release: xApp confiável que detém o privilégio exclusivo de descriptografar dados. Conforme apresentado no Algoritmo 3, este xApp recebe a chave privada da rApp de Gerenciamento de Chaves e consulta periodicamente a SDL em busca de resultados sinalizados para descriptografia. Com a chave privada, este xApp descriptografa a predição criptografada produzida pela xApp de Inferência para obter o resultado final em

Algoritmo 2 Inference xApp

Init
  Load FHE server model
  Receive the public key $public\_key$ from rApp/SMO
  Connect to SDL Database
EndInit
MainLoop
  $encrypted\_data \leftarrow$ A1-EI Interface
  if $encrypted\_data$ **then**
    $encrypted\_output \leftarrow P(public\_key, encrypted\_data)$
    Update SDL record with the $encrypted\_output$
  end if
EndMainLoop

Algoritmo 3 Data Release xApp

Init
  Load FHE client model
  Connect to SDL Database
  Receive the private key $private\_ke$ through rApp/SMO
EndInit
MainLoop
  $encrypted\_prediction \leftarrow$ Query SDL for prediction data flagged for decryption
  if $encrypted\_prediction$ **then**
    $result \leftarrow D(private\_key, encrypted\_prediction)$
    Update record in SDL with plain text prediction
    Flag row for xApp control action
  end if
EndMainLoop

texto claro (por exemplo, "anomalia") e atualiza o registro correspondente no banco de dados.

xApp RC Slice Control: Este xApp (Algoritmo 4) é responsável por executar ações com base nos resultados do modelo. Primeiramente, os resultados em texto claro gerados pelo xApp Data Release são obtidos do SDL e agregados por UE para identificar ameaças persistentes. Com base na taxa de anomalias calculada para cada *slice* (ou UE), a alocação de largura de banda e a cota de *Physical Resource Blocks* (PRBs) são ajustadas. Caso um usuário malicioso seja confirmado, o xApp revoga completamente sua largura de banda (define PRB como 0%) e aciona um comando de *Radio Resource Control* (RRC) para desconectar o usuário malicioso da rede. Este xApp aplica estas políticas de largura de banda enviando mensagens de controle E2 padronizadas para os nós E2. Este xApp de tratamento de anomalias é apresentado em [Tsourdinis et al. 2024].

É uma preocupação válida que a xApp de Inferência, embora incapaz de descriptografar o resultado, ainda possa obter informações ao observar a predição final em texto claro após a xApp confiável Data Release gravá-la no SDL. Felizmente, esta vulnerabilidade pode ser tratada por meio de um esquema adequado de autorização. Ao utilizar o Controle de Acesso Baseado em Papéis (*Role-Based Access Control* – RBAC) do SDL, as permissões do xApp de Inferência podem ser limitadas à escrita de sua saída criptografada, sendo explicitamente proibido o acesso de leitura ao campo separado em texto claro preenchido pelo xApp confiável Data Release. Além disso, como uma segunda camada de

Algoritmo 4 RC Slice Control xApp

```
Init
    Connect to RT-RIC and subscribe to the Ran Control Service Model(RC SM).
    Initialize UE data structure with default values.
    Connect to SDL Database
EndInit
MainLoop
    while True do
        Query the SDL for the data of the last 30 packet's
        Aggregate the last 30 packets per S-NSSAI
        Parse the result to extract UE ID, S-NSSAI, and anomaly ratios
        Update UE data structure
        Determine PRB allocations based on anomaly ratios.
        if a UE is an attacker (100% anomaly ratio) then
            Set PRB allocation to 0
            Distribute remaining PRB among other UEs
            Trigger RRC release for the attacker UE
        else
            Adjust PRB allocations to ensure total does not exceed 100%
            Apply slicing changes to enforce PRB allocations
        end if
    end while
EndMainLoop
```

proteção, a natureza não determinística do esquema FHE oferece proteção inerente, i.e., mesmo que o xApp pudesse visualizar o resultado, esta propriedade impede a construção de um dicionário malicioso que mapeie textos cifrados para seus respectivos resultados.

Uma consideração crítica nesta arquitetura é o requisito de latência do laço de controle do Near-RT RIC, que varia entre 10 milissegundos e 1 segundo. Como o tempo de inferência com FHE pode se tornar um gargalo significativo, a solução se baseia em um modelo de comunicação assíncrona por meio do SDL. Além disso, a complexidade do modelo de aprendizado de máquina (por exemplo, a profundidade e o número de árvores do RF) utilizado no xApp de inferência deve ser cuidadosamente escolhida para garantir que a latência do laço de controle permaneça dentro dos limites aceitáveis. Finalmente, a tarefa computacionalmente intensiva de treinamento do modelo é normalmente realizada de forma offline no Non-RT RIC, que opera em uma escala de tempo superior a um segundo. O modelo treinado resultante é então implantado no Near-RT RIC como um xApp, onde faz a inferência em tempo real (menos de 1 segundo).

5. Experimentos

Esta seção apresenta a avaliação experimental da solução proposta, com o objetivo de responder às seguintes questões principais: *É viável aplicar a solução proposta baseada em FHE para detecção de anomalias na arquitetura O-RAN? Como a solução proposta se comporta, em termos de métricas de desempenho preditivo (e.g., acurácia e precisão), quando comparada a uma solução sem preservação de privacidade? Qual é o tempo de inferência da abordagem proposta e como ele se compara ao da solução sem privacidade?* De forma geral, os resultados experimentais indicam a existência de um *trade-off* fundamental entre o tempo de predição e o desempenho preditivo.

5.1. Datasets

Os experimentos utilizam dois datasets, KDD CUP 1999 e NF3-UNSW-NB15, que permitem uma avaliação padronizada e comparação com outros trabalhos que também os utilizam [Tsourdinis et al. 2024].

KDD CUP 1999. Criado para a Terceira Competição Internacional de Descoberta de Conhecimento na KDD-99 [Obeidat et al. 2018]. O objetivo era desenvolver um modelo preditivo capaz de diferenciar entre intrusões de rede e conexões normais. Ele inclui um conjunto padrão de dados de auditoria com intrusões simuladas dentro de um ambiente de rede militar.

NF3-UNSW-NB15. Desenvolvido pelo ACCS Cyber Range Lab, o dataset oferece um formato NetFlow padronizado, enriquecido com características temporais e 12 cenários de ataque sintéticos para a avaliação de modelos de ML [Luay et al. 2025]. Contém mais de 2,3 milhões de fluxos, dos quais 5,40% são maliciosos e 94,60% são benignos.

5.2. Pré-processamento de Dados

Para o xApp de detecção de anomalias apresentado neste trabalho, as características (*features*) que podem comprometer a privacidade são: `service`, `src_bytes`, `dst_bytes` e `protocol_type`. A característica `service` identifica diretamente a aplicação ou atividade do usuário, como navegação na Web ou chamadas, permitindo a criação de perfis detalhados de uso. As características `src_bytes` e `dst_bytes` são igualmente importantes, uma vez que quantificam o volume de dados transmitidos, podendo revelar padrões de uso e consumo de recursos. Por fim, `protocol_type` indica o tipo de comunicação. Na solução sem privacidade, o xApp de terceiros tem acesso direto a todas essas informações. Por outro lado, na solução proposta, o rApp Data Lock criptografa esses dados antes de enviá-los ao xApp de Inferência de terceiros.

Com base nessa observação, foi desenvolvido um pipeline de pré-processamento para preparar os datasets tanto para o treinamento de modelos tradicionais quanto para modelos baseados em FHE. Para todos os datasets, foi usado um `ColumnTransformer` para lidar com diferentes tipos de dados. As características numéricas foram normalizadas para o intervalo $[0, 1]$ utilizando o `MinMaxScaler`, enquanto as características categóricas, como `service` e `protocol_type`, foram convertidas para um formato numérico por meio do `OneHotEncoder`. O codificador foi configurado para ignorar categorias desconhecidas encontradas durante a fase de teste.

Para o dataset NF3-UNSW-NB15, o processo de one-hot encoding resultou em um espaço de características de dimensionalidade extremamente alta, o que impacta significativamente no tempo de inferência. Para mitigar esse problema, foi aplicada a Decomposição em Valores Singulares (*Singular Value Decomposition* – SVD) como técnica de redução de dimensionalidade no NF3-UNSW-NB15 [Oseledets and Tyrtyshnikov 2009]. O número de componentes da SVD foi selecionado empiricamente para 100, com o objetivo de alcançar um equilíbrio prático entre a redução da carga computacional e a preservação da acurácia preditiva do modelo.

O tempo médio para aplicar todo esse pipeline de pré-processamento (normalização, codificação e SVD, quando aplicável) e, em seguida, realizar a quantização e cifrar com FHE uma única amostra foi de 0,0198 segundos.

5.3. Inferência baseada em FHE

Desde o inicio, o principal desafio da proposta foi a elevada latência de inferência imposta pelo FHE. Para as técnicas baseadas em FHE, foi utillizada a biblioteca Concrete-ML [Zama 2022], enquanto que para a solução sem privacidade utilizou-se a biblioteca scikit-learn [Pedregosa et al. 2011]. Uma implementação usando valores padrões para os hiper-parâmetros, como em [Tsourdinis et al. 2024], não é prática pois, como mostrado adiante, além de demandar muito tempo de processamento não apresenta os melhores resultados preditivos. Consequentemente, nos experimentos foi avaliado o impacto de dois hiper-parâmetros principais que governam diretamente a complexidade do circuito FHE: `n_estimators` - o número de árvores no ensemble - o tempo total de inferência em FHE cresce linearmente com esse parâmetro, uma vez que o circuito precisa avaliar cada árvore; e `max_depth` - a profundidade máxima de cada árvore - este é o fator mais crítico, pois mesmo um pequeno aumento na profundidade pode levar a um crescimento significativo na complexidade do circuito FHE.

Foi aplicada uma exploração metódica e foram testados os valores de `n_estimators` de 2 e 4, e valores de `max_depth` de 2 e 4. Também foi testado um valor de `n_estimators` igual a 100 para o dataset KDD CUP, especificamente para estabelecer uma comparação com o modelo sem privacidade apresentado em [Tsourdinis et al. 2024]. Os testes nessas configurações, procuraram identificar um ponto de equilíbrio que concilia alta acurácia preditiva com um tempo de inferência aceitável para a arquitetura O-RAN.

5.4. Configuração Experimental e Metodologia

A metodologia experimental foi estruturada em duas fases: Primeiro, determinar se a inferência baseada em FHE era computacionalmente viável. Assim, inicialmente foi avaliada a sobrecarga computacional imposta pela FHE. Para isso, o modelo de inferência foi executado e as métricas de desempenho foram coletadas em um ambiente Python isolado. Essa escolha metodológica foi essencial para isolar o tempo de processamento do Concrete-ML de quaisquer outras variáveis, como latência de rede ou sobrecargas específicas do O-RAN. Embora os resultados tenham demonstrado que a inferência em tempo sub-segundo é tecnicamente alcançável, esse nível de desempenho oferece pouca margem para variabilidade ou erro. Esse achado confirmou que uma arquitetura assíncrona baseada no SDL, que desacopla a inferência de alta latência do laço de controle em tempo real do Near-RT RIC, é a alternativa mais viável.

Na segunda fase foi implementada e validada a arquitetura funcional em um *testbed* O-RAN, com base na arquitetura apresentada em [Tsourdinis et al. 2024]. Todos os experimentos foram conduzidos em um sistema equipado com um processador Intel Cascade Lake de 3,9 GHz (8 vCPUs) e 32 GB de memória RAM, executando Debian GNU/Linux 12 (bookworm). O testbed 5G O-RAN utilizou o OpenAirInterface [Kaltenberger et al. 2019] para implementar o 5G Core, a RAN, e o equipamento de usuário (UE). Também foi utilizado o FlexRIC [Schmidt et al. 2021] para implementar o Near-RT RIC. No testbed foi validada com sucesso a implementação funcional de todo o pipeline de controle da solução.

5.5. Resultados Experimentais

O desempenho dos modelos com e sem privacidade é detalhado nas Tabelas 1 e 2. Nas tabelas, o tempo de inferência mede a execução da função de predição do modelo em um

Tabela 1. KDD CUP 1999

Modelo	n_estimators	max_depth	Accuracy	Precision	Recall	F1-Score	Tempo de Inferência (s)
Baseline (sem privaciadade)	2	2	0.8129	0.7932	0.9080	0.8468	0.0003
		4	0.8682	0.9437	0.8172	0.8759	0.0003
	4	2	0.8129	0.7932	0.9080	0.8468	0.0004
		4	0.8522	0.9016	0.8310	0.8648	0.0004
	100	2	0.8522	0.9016	0.8310	0.8648	0.0027
		4	0.8615	0.8830	0.8724	0.8776	0.0027
		None	0.7816	0.8931	0.7001	0.7850	0.0028
Nossa proposta (com privacidade)	2	2	0.8129	0.7932	0.9080	0.8468	0.8871
		4	0.8682	0.9437	0.8172	0.8759	1.2094
	4	2	0.8129	0.7932	0.9080	0.8468	1.0274
		4	0.8522	0.9016	0.8310	0.8648	1.6074
	100	2	0.8522	0.9016	0.8310	0.8648	7.4552
		4	0.8615	0.8830	0.8724	0.8776	18.7340
		None	0.7816	0.8931	0.7001	0.7850	2141.8062

Tabela 2. NF3-UNSW-NB15 com 100 componentes SVD

Modelo	n_estimators	max_depth	Accuracy	Precision	Recall	F1-Score	Tempo de Inferência (s)
Baseline (sem privacidade)	2	2	0.9619	0.9592	0.9619	0.9464	0.0010
		4	0.9730	0.9728	0.9730	0.9670	0.0010
	4	2	0.9628	0.9603	0.9628	0.9483	0.0010
		4	0.9730	0.9727	0.9730	0.9670	0.0010
Nossa proposta (com privacidade)	2	2	0.9619	0.9592	0.9619	0.9464	0.8062
		4	0.9730	0.9728	0.9730	0.9670	1.4223
	4	2	0.9628	0.9603	0.9628	0.9483	0.9508
		4	0.9730	0.9727	0.9730	0.9670	2.0239

único fluxo de rede, sem o pré-processamento e a cifragem dos dados, que conforme já relatado levam 0,0198 segundos. Um dos principais resultados é a viabilidade prática da abordagem proposta para privacidade, pois para ambos datasets foram obtidos tempos de inferência inferior a um segundo. A viabilidade da solução é ainda reforçada pelo modelo de comunicação assíncrona do SDL, desacoplando a computação FHE das demais tarefas do loop de controle, garantindo que a plataforma permaneça responsiva e não bloqueie outras funções dos xApps. A abordagem assíncrona oferece a flexibilidade necessária e permite que o sistema lide de forma robusta tanto com modelos de inferência abaixo de um segundo quanto com fluxos de trabalho mais complexos.

Um segundo resultado crucial é a preservação perfeita do desempenho preditivo. Todos os modelos com FHE produziram métricas de predição idênticas em relação às suas contrapartes sem privacidade, o que é um benefício direto do esquema TFHE implementado pelo Concrete-ML, projetado para executar exatamente a lógica das árvores de decisão, sem recorrer a aproximações. No entanto, embora estas métricas sejam preservadas, a relação entre complexidade do modelo e tempo de predição apresenta um desafio crítico. Enquanto o aumento do número de estimadores ou da profundidade das árvores resulta em um acréscimo de tempo desprezível para modelos sem privacidade, existe uma relação não linear nos modelos com privacidade. Por exemplo, no conjunto de dados KDD CUP, a mudança de uma configuração mais simples (2 estimadores, profundidade 2) para uma configuração mais complexa (100 estimadores, profundidade 4) aumenta o tempo de inferência de 0,88 segundos para 18,73 segundos — um aumento de latência de aproximadamente $21\times$. Entretanto, este aumento no tempo computacional não produz um retorno justificável em desempenho preditivo e, em alguns casos, faz com que o modelo se torne pior. Por exemplo, nos resultados do KDD CUP, um modelo mais simples (2 estimadores, profundidade 4) alcança uma acurácia de 0,8682 em 1,2 segundos. Em contraste, um modelo mais complexo (100 estimadores, profundidade 4) resulta em uma

acurácia inferior de 0,8615 e ainda requer 18,73 segundos.

6. Conclusões e Trabalhos Futuros

Este artigo apresentou uma arquitetura para detecção de anomalias com preservação de privacidade no contexto do O-RAN, que enfrenta os riscos de privacidade introduzidos pelo acesso de xApps de terceiros a dados sensíveis da rede e de usuários. A avaliação experimental mostra duas conclusões principais. Primeiro, os modelos com FHE alcançaram desempenho preditivo idêntico às suas contrapartes em texto claro, confirmando que privacidade pode ser implementada sem qualquer degradação na acurácia do modelo. Segundo, a sobrecarga computacional dos modelos que utilizam FHE pode ser gerenciada com sucesso. Enquanto um modelo implementado de forma ingênua mostrou-se inviável (mais de 2141 segundos para inferência), existe um *trade-off* prático entre complexidade do modelo e latência, i.e., ao ajustar os hiperparâmetros foi possível alcançar tempos de inferência inferiores a um segundo. Esse desempenho, aliado ao modelo de comunicação assíncrona viabilizado pelo SDL, torna a solução compatível com os limites operacionais do laço de controle do Near-RT RIC.

Os princípios demonstrados neste artigo podem ser estendidos para outros casos de uso e no desenvolvimento de xApps com preservação de privacidade para gerenciamento inteligente de recursos. A aplicação destas técnicas a um conjunto mais amplo de funções de rede é fundamental para aprimorar a segurança e a confiabilidade da O-RAN.

Referências

Acar, A., Aksu, H., Uluagac, A. S., and Conti, M. (2018). A survey on homomorphic encryption schemes: Theory and implementation. *ACM Comput. Surv.*, 51(4).

Cheon, J. H., Kim, A., Kim, M., and Song, Y. (2016). Homomorphic encryption for arithmetic of approximate numbers. Cryptology ePrint Archive, Paper 2016/421.

Chiejina, A., Kim, B., Chowhdury, K., and Shah, V. K. (2024). System-level analysis of adversarial attacks and defenses on intelligence in o-ran based cellular networks. In *17th ACM Conference on Security and Privacy in Wireless and Mobile Networks*.

Dias, V., Silva, M., Gomes, M., Oliveira, L., Riker, A., and Abelém, A. (2024). Mitigação inteligente de ataques DDoS em redes o-RAN utilizando aprendizado de máquina. In *XV Workshop de Pesquisa Experimental da Internet do Futuro*, pages 55–62. SBC.

Dryjanski, M., Kulacz, L., and Kliks, A. (2021). Toward modular and flexible open ran implementations in 6g networks: Traffic steering use case and o-ran xapps. *Sensors*.

Fan, J. and Vercauteren, F. (2012). Somewhat practical fully homomorphic encryption. Cryptology ePrint Archive, Paper 2012/144.

Frery, J., Stoian, A., Bredehoft, R., Montero, L., Kherfallah, C., Chevallier-Mames, B., and Meyre, A. (2023). Privacy-preserving tree-based inference with fully homomorphic encryption. Published: Cryptology ePrint Archive, Paper 2023/258.

Grinsztajn, L., Oyallon, E., and Varoquaux, G. (2022). Why do tree-based models still outperform deep learning on typical tabular data? *Advances in neural information processing systems*, 35:507–520.

Kaltenberger, F., De Souza, G., Knopp, R., and Wang, H. (2019). The OpenAirInterface 5g new radio implementation: Current status and roadmap. In *23rd ITG Workshop on Smart Antennas, Demo Session*.

Khan, M. S., Farzaneh, B., Shahriar, N., Saha, N., and Boutaba, R. (2022). SliceSecure: Impact and detection of DoS/DDoS attacks on 5g network slices. In *2022 IEEE Future Networks World Forum (FNWF)*, pages 639–642. IEEE Computer Society.

Lacava, A., Polese, M., Sivaraj, R., Soundrarajan, R., Bhati, B. S., Singh, T., Zugno, T., Cuomo, F., and Melodia, T. (2023). Programmable and customized intelligence for traffic steering in 5g networks using open ran architectures. *IEEE Transactions on Mobile Computing*, 23(4):2882–2897.

Lin, D., Bhargav, S., Chiejina, A., Ibrahem, M. I., and Shah, V. K. (2025). Zt-ric: A zero trust ric framework for ensuring data privacy and confidentiality in open ran. In *IEEE 22nd Consumer Communications & Networking Conference (CCNC)*.

Louppe, G. (2014). *Understanding random forests: From theory to practice*. Universite de Liege (Belgium).

Luay, M., Layeghy, S., Hosseininoorbin, S., Sarhan, M., Moustafa, N., and Portmann, M. (2025). Temporal analysis of NetFlow datasets for network intrusion detection systems. In *https://arxiv.org/abs/2503.04404*.

O-RAN ALLIANCE (2022). O-RAN.WG1.O-RAN-Architecture-Description-v07.00. Technical Specification 7, O-RAN ALLIANCE.

Obeidat, I., Hamadneh, N., Al-kasassbeh, M., and Almseidin, M. (2018). Intensive pre-processing of KDD cup 99 for network intrusion classification using machine learning techniques. *arXiv preprint arXiv:1805.10458*.

Oseledets, I. V. and Tyrtyshnikov, E. E. (2009). Breaking the curse of dimensionality, or how to use svd in many dimensions. *SIAM Journal on Scientific Computing*, 31(5):3744–3759.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12.

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.

Schmidt, R., Irazabal, M., and Nikaein, N. (2021). FlexRIC: an SDK for next-generation SD-RANs. In *Proceedings of the 17th International Conference on Emerging Networking EXperiments and Technologies*, CoNEXT '21, pages 411–425.

Tsourdinis, T., Makris, N., Korakis, T., and Fdida, S. (2024). AI-driven network intrusion detection and resource allocation in real-world o-RAN 5g networks. In *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking*.

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*.

Zama (2022). Concrete ML: a privacy-preserving machine learning library using fully homomorphic encryption for data scientists. In *https://docs.zama.org/concrete-ml*.