Um Sistema de Detecção de Ataques *Evil Twin* com Aprendizado de Máquina Não-Supervisionado

Ricardo L. Cerqueira Júnior[1], **Felipe da R. Henriques**[1], **Igor M. Moraes** [2], **Dalbert M. Mascarenhas**[1]

[1]Centro Federal de Educação Tecnológica Celso Suckow da Fonseca - CEFET/RJ
Petrópolis - RJ - Brasil

[2]Laboratório MidiaCom – IC/TCC/PGC
Universidade Federal Fluminense (UFF), Niterói – RJ – Brasil

{felipe.henriques,dalbert.mascarenhas}@cefet-rj.br,
ricardo.junior@aluno.cefet-rj.br, igor@ic.uff.br

Abstract. *This paper proposes a User-based Evil Twin Attacks Detection System that employs One Class Supporting Vector Machine for anomaly detection in IEEE 802.11 wireless networks. The proposed system is user-centric and uses user device interactions with access points to detect the attack. Evil Twin attacks are replicated experimentally to create two datasets that are used to train and refine the proposed system: one with data from legitimate access points only and the other also with data from malicious access points. The results show the high effectiveness of the proposed system, with an accuracy rate of 98.72% while maintaining sensitivity around 90%, thus demonstrating the proposed system's ability to detect Evil Twin attacks.*

Resumo. *Este artigo propõe um Sistema de Detecção de Ataques* Evil Twin *Baseado no Usuário, que usa a Máquina de Vetor de Suporte de Uma Classe (OCSVM) para detecção de anomalias em redes sem fio IEEE 802.11. O sistema proposto é centrado no usuário e usa as interações do dispositivo do usuário com pontos de acesso para detectar o ataque. Ataques Evil Twin são replicados experimentalmente para criar dois conjuntos de dados que são usados para treinar e refinar o sistema proposto: um somente com dados de pontos de acesso legítimos e outro também com dados de pontos de acesso maliciosos. Os resultados mostram a alta eficácia do sistema proposto, com uma taxa de precisão de 98,72% enquanto mantém a sensibilidade em torno de 90%, demonstrando, assim, a capacidade do sistema proposto de detectar ataques* Evil Twin.

1. Introdução

O ataque *Evil Twin* explora a relação entre um cliente e um ponto de acesso (*Access Point* - AP) em uma rede sem fio IEEE 802.11. No *Evil Twin*, um atacante se passa por um ponto de acesso legítimo, podendo falsificar o identificador da rede (*Service Set Identifier* - SSID) e o endereço MAC (*Basic Service Set Identifier* - BSSID) desse ponto de acesso legítimo. Assim, o atacante imita um AP legítimo e engana os clientes, fazendo com que eles se associem a um AP malicioso. Uma vez estabelecida essa associação, o atacante pode interceptar e manipular o tráfego de dados dos clientes, abrindo caminho

para atividades maliciosas adicionais. Esse tipo de ataque facilita ataques de homem-no-meio (*man-in-the-middle*), incluindo DNS *Spoofing* e falsificação de certificados, que podem levar a violações de segurança [Muthalagu e Sanjay, 2021, Faria et al., 2020].

Este artigo propõe o *User-based Evil Twin Attacks Detection System* (UETADS), um sistema de detecção de ataques *Evil Twin* baseado em aprendizado de máquina não-supervisionado. O sistema proposto emprega o *One-Class Support Vector Machine* (OCSVM) para detecção de anomalias. Por isso, o UETADS usa poucas amostras de dados para definir um padrão de atividade da rede, possibilitando uma identificação proativa em tempo real de atividades maliciosas. O sistema proposto é executado diretamente no cliente e fornece uma análise granular dos dados trocados entre o usuário e o AP ao qual ele está associado em tempo de execução. O objetivo é identificar padrões anômalos, como mudanças inesperadas em endereços físicos ou tempos de resposta incomuns, que são indicativos de potenciais ameaças à segurança do cliente. Diferente de métodos tradicionais, que dependem de análise extensiva de tráfego, o UETADS é centrado no usuário, e se concentra na interação entre cliente e o AP ao qual está associado. Essa abordagem possibilita adaptabilidade à ferramenta, criando uma padrão de comportamento considerado normal para cada rede a qual o cliente se conecte.

Experimentos práticos são realizados para identificar a mecânica dos ataques e a resposta típica dos clientes sob o ataque *Evil Twin* e também para construir os conjuntos de dados de treinamento e validação do sistema proposto. Este conjunto de dados se tornou a base para avaliar a capacidade da ferramenta de detectar especificamente anomalias. Além disso, foi desenvolvido um segundo conjunto de dados cobrindo uma variedade de cenários de ataque Evil Twin para avaliar minuciosamente a sensibilidade da ferramenta. Os hiperparâmetros do OCSVM foram ajustados após cada iteração de treinamento e teste. Esse processo foi crucial para alcançar a precisão e eficiência ótimas. Os resultados mostram a alta eficácia do sistema proposto, com uma taxa de precisão de 98,72% enquanto mantém a sensibilidade em torno de 90%, demonstrando, assim, a capacidade do sistema proposto de detectar ataques *Evil Twin*.

O restante deste artigo está organizado da seguinte forma. A Seção 2 descreve o ataque *Evil Twin*. A Seção 3 discute os trabalhos relacionados. A Seção 4 relata os experimentos práticos realizados para caracterizar o ataque e para construir os conjuntos de dados. A Seção 5 introduz o sistema proposto. A Seção 6 discute os resultados obtidos. A Seção 7 conclui este trabalho.

2. O Ataque *Evil Twin*

No ataque *Evil Twin*, um atacante se passa por um ponto de acesso legítimo IEEE 802.11, falsificando o SSID da rede e o endereço MAC desse ponto de acesso legítimo e até mesmo usando o mesmo padrão de segurança. Dessa forma, o atacante imita um AP legítimo e engana os clientes, fazendo com que eles se associem a um AP malicioso. Em geral, o AP malicioso é estrategicamente posicionado em uma posição mais próxima do cliente alvo do que a posição do AP legítimo, aumentando a probabilidade do cliente se associar ao AP malicioso porque a intensidade do sinal recebido pelo cliente e transmitido pelo AP malicioso tende a ser maior do que a intensidade do sinal recebido pelo cliente e transmitido pelo AP legítimo. Normalmente, os clientes usam a intensidade do sinal recebido para decidir com qual AP se associar. O ataque *Evil Twin* é particular-

mente eficiente quando o cliente alvo já está associado a um AP legítimo, pois o atacante pode aprimorar o ataque *Evil Twin* com uma estratégia de "desautenticação", compelindo o alvo a se conectar ao AP malicioso. A Figura1 ilustra um exemplo de ataque *Evil Twin*.

Figura 1. Um exemplo de ataque *Evil Twin*. O smartphone é o cliente e, nesse caso, tende a se associar ao Ponto de Acesso Falso, pois recebe um sinal com maior intensidade desse AP que está mais próximo fisicamente a ele do que o Ponto de Acesso Legítimo. O SSID e o endereço MAC usados pelo Ponto de Acesso Falso são os mesmos usados pelo Ponto de Acesso Legítimo. Nesse exemplo, o Atacante pode interceptar e manipular o tráfego de dados do cliente para e vindo da Internet.

O ataque *Evil Twin* pode ser executado mesmo em redes sem fio que usem a emenda 802.11i, também conhecida como Wi-Fi Protected Access 2 (WPA2). O WPA2 usa o algoritmo criptográfico AES (*Advanced Encryption Standard*) e um processo de autenticação em quatro vias entre o cliente e o AP. Durante esse processo de autenticação mútua, cria-se uma chave compartilhada entre o cliente e o AP, que não é divulgada para terceiros. No entanto, novas emendas a padrão IEEE 802.11 introduzem vulnerabilidades. A emenda IEEE 802.11r, por exemplo, trata do *roaming* e torna a troca de pontos de acesso por um cliente mais rápida, porém inclui potenciais alterações ou omissões nas etapas da autenticação mútua em quatro vias em algumas implementações.

O uso de Sistemas de Detecção de Intrusão (*Intrusion Detection Systems* - IDS) tem sido proposto para detecetar ataques *Evil Twin*. Tais sistemas são projetados para detectar ações não autorizadas que ameacem a confidencialidade, integridade ou disponibilidade de um sistema. Além disso, empregam ferramentas e mecanismos especializados para identificar anomalias no tráfego da rede ou comportamentos incomuns em aplicações e serviços. Isso é particularmente relevante diante de estratégias de ataque diversas e em evolução, como o *Evil Twin*.

3. Trabalhos Relacionados

Existem diferentes trabalhos na literatura para detectar ataques *Evil Twin* que monitoram o tráfego da rede por meio da captura de pacotes e outras fontes de dados e outros trabalhos que dependem de dados gerados dentro do cliente, como logs de serviços e dados de *firewall*. Além disso, o uso do aprendizado de máquina pode aumentar a eficácia na detecção de ataques. Ao modelar o comportamento padrão, não anômalo, da rede ou do sistema, os algoritmos de aprendizado de máquina podem identificar desvios significativos nos padrões de dados, marcando-os como potenciais ameaças.

Nakhila *et al.* descrevem um modelo baseado em varredura aleatória de canais para a detecção de ataques *Evil Twin* [Nakhila et al., 2015]. Esta abordagem envolve a

varredura de todos os pontos de acesso próximos, catalogando endereços MAC e identificadores, e armazenando-os em um servidor remoto. Esta ferramenta não requer treinamento ou uma análise aprofundada das características da rede. O programa do lado do cliente observa quadros enviados pelo servidor, procurando por endereços MAC de destino. Se um quadro exibir um endereço MAC diferente do endereço previamente coletado, a ferramenta identifica essa discrepância como indicativa de um ataque *Evil Twin*.

Kitisriworapan *et al.* propõem uma solução móvel envolvendo movimento físico e medições de tempo de ida e volta (*Round-Trip Time* - RTT) [Kitisriworapan et al., 2020]. O projeto é dividido em fases de coleta e classificação de dados. Primeiro, um algoritmo coleta valores de RTT de diferentes posições para um AP fixo. Um segundo algoritmo calcula a distância desses valores para centróides usando k-means. Se a distância exceder um limiar, suspeita-se de um ataque *Evil Twin*. Caso contrário, uma função de distribuição cumulativa é usada para investigações mais profundas.

Swetha e Shailaja exploram o uso de modelos como o Perceptron Multicamadas (MLP), Vizinho Mais Próximo (K-NN) e Árvore de Decisão para classificar potenciais ataques de pontos de acesso não autorizados em redes sem fio IEEE 802.11 [Swetha e Shailaja, 2020]. Tendo o RTT como base para a identificação do ataque, os autores relataram maior precisão com MLP (98%) e Árvore de Decisão (98,01%).

Tian *et al.* introduzem uma ferramenta de detecção baseada em Rede Neural Convolucional que usa um dispositivo para observar pontos de acesso e capturar informações relevantes como endereços MAC [Tian et al., 2021]. Essas impressões digitais da rede são usadas para treinar o modelo, que identifica ataques com alta precisão em testes preliminares.

Muthalagu e Sanjay adotam uma abordagem de contra-ataque para lidar com ataques *Evil Twin* [Muthalagu e Sanjay, 2021] e desenvolvem um IDS centralizado que monitora a rede, varrendo todos os canais por sinais SSID (BSSID) e comparando-os com uma lista branca. Além disso, busca por pacotes de desautenticação, uma característica desse tipo de ataque. Se tais sinais ou BSSIDs não listados forem encontrados, o software contra-ataca emitindo sinais de desautenticação, forçando os clientes a se desconectarem do AP malicisoso.

Mahfouz *et al.* desenvolvem um sistema de detecção de intrusão baseado em rede usando o One-Class Support Vector Machine (OCSVM) [Mahfouz et al., 2021] que aprende o comportamento da rede de maneira não supervisionada e detecta ataques com uma precisão média de 97,61%. O conjunto de dados da ferramenta para treinamento foi criado usando vários ambientes de nuvem.

Yang *et al.* implementam uma variante do OCSVM para detecção de anomalias em dispositivos de Internet das Coisas [Yang et al., 2021]. Abordando o desafio de lidar com grandes volumes de dados e custos computacionais, eles usam um Modelo de Mistura Gaussiana para seleção de recursos durante o treinamento, o que efetivamente reduz o tempo de processamento.

Hsu *et al.* usam as características de encaminhamento de pacotes do IP para detectar ataques [Hsu et al., 2022]. Projetado para implantação do lado do cliente como um IDS baseado em hospedeiro, esta ferramenta alcança alta precisão aproveitando a RSSI

(*Received Signal Strength Indication*). Embora seus resultados sejam comparáveis aos de modelos mais complexos, o autor não detalha o desempenho da ferramenta em situações de RSSI abaixo de 45%.

Wang *et al.* propõem um IDS usando One-Class SVM combinado com um Modelo de Mistura Gaussiana (GMM) em uma abordagem de aprendizado semi-supervisionado [Wang et al., 2023]. Os autores usam dois conjuntos de dados públicos para desenvolvimento, alcançando um F1-score acima de 95%, mesmo com diferentes amostras de ataques durante os testes.

Ao contrário dos trabalhos relacionados que frequentemente usam soluções centralizadas focadas em características de protocolo ou análise ampla da rede, o sistema proposto neste artigo adota uma abordagem centrada no usuário especificamente para detecção de anomalias e executa no cliente. Uma característica distintiva do UETADS é sua dependência de uma pequena quantidade de dados, coletados no início da conexão, para determinar um padrão de comportamento normal da rede. A seleção do OCSVM é fundamental para garantir a adaptabilidade em diversas redes IEEE 802.11. Este modelo de aprendizado não supervisionado requer exclusivamente dados da classe dominante para treinamento, alinhando-se com a abordagem do UETADS.

4. Experimentos com o Ataque *Evil Twin*

Experimentos práticos são realizados para identificar a mecânica dos ataques e a resposta típica dos clientes sob o ataque *Evil Twin* e também para construir os conjuntos de dados de treinamento e validação do sistema proposto. São usados APs da marca Trendnet e notebooks para desempenhar a função de AP. Os notebooks, atuando como APs, estavam conectados à Internet por meio de um cabo de rede, e suas interfaces de rede sem fio são usadas para prover associação à rede sem fio para os clientes. Os detalhes sobre os equipamentos usados são os seguintes: (i) clientes são *smartphones* Samsung Galaxy S10+, com Android 11 e notebooks Dell Inspiron 7560; (ii) pontos de acesso legítimos são APs Trendnet protegidos por senha, com WPA2-PSK e frequência de rede de 2,4 GHz; e (iii) ponto de acesso malicioso é um notebook Dell Inspiron 7560 com sistema operacional Ubuntu 20.04.

São usados os softwares gratuitos hostapd [Floeter, 2004] e dnsmasq [Kelley, 2022] para configurar o AP malicioso e executar o ataque *Evil Twin*. O hostapd cria efetivamente o AP e o dnsmasq atua como servidor DNS e DHCP. Alguns do parâmetros definidos no hostapd e dnsmasq são (i) interface, que define o nome da interface de rede que será usada como um AP; (ii) SSID, que é o identificador da rede idêntico ao SSID do AP legítimo; (iii) DHCP_SERVER, que indica o endereço do servidor DHCP, que nesse caso, é o endereço de *loopback* por considerar que esse serviço é hospedado na mesma máquina; e (iv) AUTH_ALGS, que define se a rede é protegida por senha ou aberta. Como o foco deste experimento é o estudo do ataque *Evil Twin*, e especificamente o comportamento da vítima durante o ataque, outros possíveis ataques não são implementados, e todo o tráfego do cliente é direcionado para a Internet.

Os equipamentos e ferramentas detalhados na Seção 4 são usados para emular o ataque *Evil Twin*, incorporando variações nas implementações de segurança. São realizados três experimentos e em todos os experimentos, as distâncias entre os dispositivos são mantidas para minimizar mudanças significativas na intensidade do sinal recebido

pelos clientes. Os *smartphones* estão a 5 m do AP malicioso e a 10 m do AP legítimo. Inicialmente, todos os clientes se associam ao AP legítimo. Mais detalhes sobre os três experimentos, incluindo as várias variações de implementação de segurança, serão apresentados a seguir.

No primeiro experimento, o AP malicioso não exige senha para autenticação dos clientes e o AP legítimo sim, uma vez que está configurado com WPA2-PSK. O AP malicioso replica apenas o SSID do AP legítimo. O ataque auxiliar de "desautenticação", executado com a ferramenta `aircrack-ng` [Aircrack-ng, 2021], leva à desconexão da vítima, mas não há associação subsequente com o AP malicioso. O insucesso do ataque decorre da verificação do sistema das características de rede salvas. Devido à diferença nos padrões de segurança entre as redes, o cliente reconhece tal diferença e não se associa ao AP malicioso, pois tal AP não possui os parâmetros de rede fornecidos anteriormente pelo AP legítimo.

No segundo experimento, executa-se um ataque típico de *Evil Twin*. O AP malicioso está configurado para se passar pelo AP legítimo com o mesmo SSID e padrão de segurança. O endereço MAC do AP legítimo não foi replicado pelo AP malicioso. O ataque auxiliar de "desautenticação", leva à desconexão imediata da vítima, para que em seguida ele se associe ao AP malicioso. Portanto, o ataque é bem sucedido.

No terceiro experimento, a configuração do AP malicioso é semelhante à anterior. Ele usa o o mesmo SSID, padrão de segurança e senha do AP legítimo. No entanto, desta vez não se usa o `aircrack-ng` para desconectar forçosamente o cliente. Em vez disso, observa-se uma troca espontânea do cliente para o AP malicioso, cuja intensidade do sinal recebido pelo cliente é maior. Em trabalhos anteriores, tal troca espontânea foi observada quando ambos os APs compartilhavam o mesmo identificador de rede [Wi-Fi Alliance, 2020], como é o caso deste experimento. Nesse caso, o tempo médio necessário para essa troca varia de 7 a 26 s.

5. O Sistema Proposto *User-based Evil Twin Attacks Detection System* (UETADS)

Algoritmos de aprendizado de máquina, como Support Vector Machine (SVM) e K-nearest neighbor(K-NN), já foram integrados em IDSes. No entanto, à medida que os padrões de ataque evoluem, como no caso do ataque *Evil Twin*, surge a necessidade de abordagens mais sofisticadas para manter a precisão da detecção. É nesse sentido que se encaixa o One-Class Support Vector Machine, demonstrando um potencial significativo na detecção de anomalias, especialmente em cenários que envolvem dados não rotulados. O OCSVM, um modelo de aprendizado não supervisionado, é particularmente apto para a detecção de anomalias. Esse modelo é especialmente projetado para identificar *outliers* nos dados, que podem significar intrusões. Diferentemente das SVMs tradicionais e dos Perceptrons Multicamadas, que operam em uma base de aprendizado supervisionado para classificação, o OCSVM é adaptado para dados de uma única classe dominante. No OCSVM, o processo de aprendizado envolve mapear pontos de dados em um hiperespaço, definindo um limite, seja um hiperplano ou uma hiperesfera, para separar dados normais (*inliers*) de anomalias (*outliers*). Este modelo calcula uma função binária que interpreta a densidade de probabilidade dos dados de entrada, utilizando um mapa de características e um espaço de produto interno. O objetivo principal é classificar novas observações com

base em seu alinhamento com a região do modelo treinado, distinguindo efetivamente entre comportamento normal e potenciais ameaças de segurança.

O sistema proposto é executado em um cliente e coleta dados necessários para o treinamento durante os primeiros minutos da associação do cliente com um ponto de acesso. Essa abordagem, aliada ao uso do OCSVM, faz com que o UETADS seja um sistema adaptável a diferentes redes Wi-Fi e clientes que, com poucos dados de treinamento, se tornam capazes de identificar um padrão de comportamento e apontar anomalias. O UETADS é desenvolvido em Python 3 e usa bibliotecas OS, NumPy, Pandas, scikit-learn e t-SNE.

5.1. Seleção de Atributos de Conexão

Ao se conectar a um AP, o cliente recebe via DHCP uma oferta e escolhe um endereço IP da sub-rede do AP para ser usado por sua interface de rede. Com essa configuração concluída, o cliente pode se comunicar com outros dispositivos através do AP. Atributos como o endereço IP e a máscara de sub-rede, uma vez configurados, não são frequentemente modificados durante a associação do cliente com o AP e, por isso, esses dois atributos são considerados pelo sistema proposto para a análise de um possível ataque *Evil Twin*.

Outra característica útil para identificar mudanças, como a inclusão de novos roteadores na rota de comunicação, é o tempo de vida (*Time to Live* - TTL) dos pacotes. Quanto maior o número de saltos em uma rota, maior a possibilidade de mudanças. No entanto, em uma rede local, ao medir o TTL para o primeiro salto após o *gateway* interno da rede, a possibilidade de mudanças é pequena. Consequentemente, essa abordagem permite considerar a variação nos endereços IP detectados pelo comando `traceroute` até o segundo salto dentro da rede local.

Dado que a introdução de um novo AP malicioso pode adicionar um novo salto na rota de comunicação e levar a um encaminhamento adicional de pacotes, podem ocorrer mudanças no tempo de resposta para o mesmo destino devido ao novo AP falso na rede. Consequentemente, o comando `ping` pode ser usado repetidamente para um destino específico para determinar os valores comuns de RTT para a rede. No caso de um ataque, espera-se que esses tempos de resposta apresentem mudanças que, quando analisados juntamente com outros dados, podem auxiliar na detecção da anomalia.

Atacantes que executam ataques *Evil Twin* comumente replicam todos os dados conhecidos do AP legítimos, para que o cliente conclua que se trata do mesmo AP ao qual se associou ou já se associou anteriormente. No entanto, conforme observado nos testes realizados na Seção 4, não é necessário configurar o AP malicioso com todas as características do AP legítimo. Portanto, o endereço IP e o endereço MAC do *gateway* da rede são atributos válidos para o UETADS verificar.

Por fim, o atacante pode optar por não realizar ataques auxiliares como a "desautenticação" e dificultar a detecção por ferramentas baseadas nessa assinatura. Conforme observado na Seção 4, o cliente pode ter a capacidade de trocar espontaneamente de AP, caso receba um sinal de maior intensidade de um outro AP. Por isso, a RSSI também é usado pelo UETADS no processo de detecção.

5.2. Detalhes de Implementação

Ao ser iniciado no cliente, o UETADS avalia os atributos atuais da rede e, usando o SSID, procura ocorrências anteriores de associações a redes identificadas pelo mesmo SSID em suas execuções históricas. Se o UETADS reconhecer o nome da rede, indicando que o SSID está presente em seu banco de dados, este recupera o modelo OCSVM armazenado e contrasta os novos valores da rede com os últimos valores registrados de sua execução anterior. Se o UETADS não encontrar uma entrada correspondente ao SSID, torna-se necessário determinar o comportamento padrão da rede e registrar as características descritas na Seção 5.1 durante um período de tempo especificado.

Na implementação do UETADS, é usado o comando `ping` para estimar o tempo de resposta para o endereço previamente designado como o *gateway* da rede. Para cada nova entrada no *dataset*, o respectivo RTT, R_{x_*}, é calculado como a média ponderada do RTT anterior, R_x e a medição recente, M, segundo a equação $R_{x_*} = \alpha \cdot R_x + (1 - \alpha) \cdot M$. Diferentes valores para α foram testados durante a implementação. O UETADS apresentou o melhor desempenho com $\alpha = 0,9$.

Uma vez construído o *dataset*, o treinamento é realizado e o UETADS inicia a análise dos novos atributos da rede. Se o OCSVM considerar que há um AP malicioso na rede, um alerta é enviado ao usuário do dispotivo cliente. Do contrário, se o algoritmo retorna que o AP é legítimo, os valores lidos são atualizados como os últimos atributos seguros x e o ciclo recomeça, como indica a Figura 2.

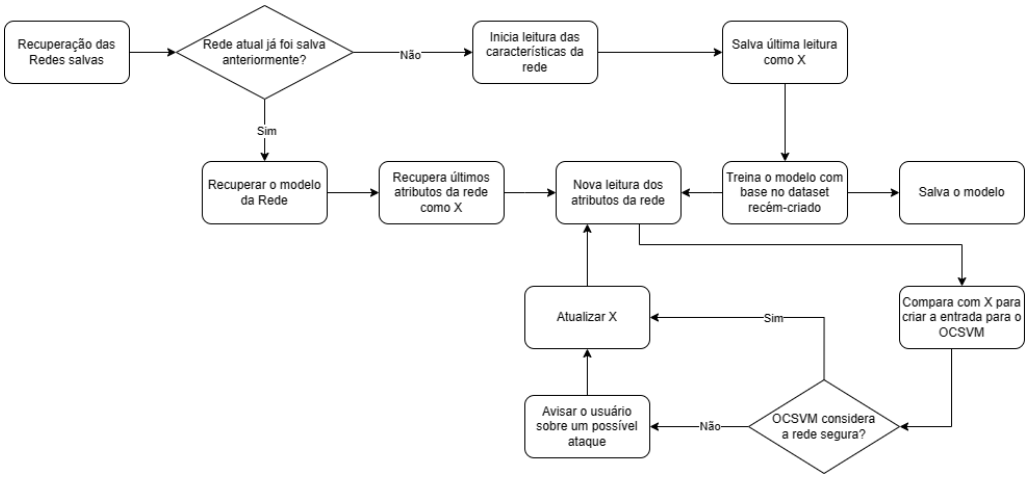

Figura 2. O fluxo de execução do sistema proposto UETADS.

Os atributos textuais como máscara de rede, IP e MAC do AP e o endereço IP do segundo salto conhecido via `traceroute` são comparados com os valores da iteração anterior x do UETADS (Algoritmo 1). Se o novo valor encontrado for igual à leitura anterior, seu respectivo campo na tupla recebe 1, e atributos diferentes recebem 0. A Tabela 1 apresenta valores usados como entrada para testar o modelo. Como uma troca de AP cria variações significativas na RSSI, um novo campo VAR_RSSI é usado para registrar variações da intensidade do sinal maiores que um determinado limiar. O valor escolhido nesta implementação foi de 8 dB, seguindo o padrão estabelecido para *roaming* em dispositivos iOS que estão transmitindo dados na rede [Apple Inc.,]. Comparando x_{*RSSI} com x_{RSSI}, variações menores que o limiar recebem 1; caso contrário, o valor atribuído é 0.

Algoritmo 1: Construção do *dataset* de treinamento.

$dataset\ d = []$
$x_* = dadosAtuais()$
while $t \geq 0$ **do**
    if $x_{*_i} = x_i, i \in \{MAC, ER, IP, TTL\}$ **then**
        $k_i = 1$
    else
        $k_i = 0$
    if $(x_{*_{RSSI}} - x_{RSSI}) < 8$ **then**
        $K_{VAR\_RSSI} = 1$
    else
        $K_{VAR\_RSSI} = 0$
    $x_{*_{RTT}} = 0.9 \cdot x_{RTT} + 0.1 \cdot M$
    $k_{RTT} = normalize(x_{*_{RTT}})$
    $k_{RSSI} = normalize(x_{*_{RSSI}})$
    d.append(k)
    $x = x_*$
    $x_* = dadosAtuais()$

Tabela 1. Conjunto de dados de teste contendo as entradas de anomalia geradas.

	MAC	NA/MASK	IP	RTT	TTL	RSSI	VAR_RSSI
0	1	0	1	28.867	0	-35	1
1	1	1	0	16.246	1	-30	1
2	1	0	0	3.140	1	-34	1
3	1	1	0	22.674	1	-32	0
...							

Após a coleta dos dados, as colunas RTT e RSSI são separadamente adequadas ao intervalo $[0,1]$, com o ajuste do modelo de normalização sendo realizado sobre o *dataset* de treino. Esse método de aproximação dos dados já se provou útil na redução do tempo de treinamento, além de melhorar a acurácia, a depender do método escolhido [Li e Liu, 2011].

5.3. Conjunto de Dados de Treinamento e Teste

Para obter dados de associações aos APs legítimos, executa-se o Algoritmo 1 para coletar dados da rede de um único AP durante 600 s, registrando a cada 0,1 s. Esse procedimento gera um arquivo com 6000 entradas, que são divididas para treinamento do modelo e testes de especificidade. Dados anômalos são coletados durante a emulação de ataques *Evil Twin*, descritos na Seção 4. Para transparência e reprodutibilidade, o código-fonte e os arquivos de entrada usados para obter os resultados da Seção 6 estão disponíveis [Cerqueira Júnior, 2023].

Com ambos os conjuntos de dados disponíveis, usa-se o Coeficiente de Correlação de Pearson para identificar relações lineares entre os atributos selecionados entre si e

com a saída do modelo, definida na Tabela 2 como y. A análise revela que o atributo VAR_RSSI (coluna) tem uma correlação mais baixa com o resultado final, mas ainda possui uma correlação linear altae de $0,496$. Além disso, observando a correlação das características de entrada, todos os coeficientes de correlação estão mais próximos de $|1|$ do que de $|0|$. Isso desencoraja a remoção de colunas da entrada. No entanto, realiza-se também um treinamento com redução na dimensão dos dados e o resultado está disponível ao final da Seção 6.

Tabela 2. Matriz de correlação reduzida.

ER/MASK	0.244						
IP	0.301	0.226					
RTT	-0.309	-0.283	-0.303				
TTL	0.261	0.244	0.257	-0.272			
RSSI	-0.332	-0.348	-0.340	0.416	-0.347		
VAR_RSSI	0.251	0.277	0.283	-0.296	0.252	-0.366	
y	0.524	0.505	0.518	-0.607	0.502	-0.689	0.496
	MAC	ER/MASK	IP	RTT	TTL	RSSI	VAR_RSSI

6. Resultados

Após o treinamento, os modelos são testados em dois conjuntos de dados: entradas legítimas não treinadas e entradas de ataques *Evil Twin*, conforme detalhado na Seção 5.3. O primeiro teste avalia a especificidade do modelo, calculando a proporção de negativos verdadeiros T_N em relação ao total de casos testados, que é dado pela soma de T_N com os falsos positivos F_P . O segundo teste avalia a sensibilidade do modelo, identificando falsos negativos F_N, indicando cenários de ataques não reconhecíveis. A acurácia geral do modelo é determinada pela proporção de detecções corretas em ambos os testes em relação ao número total de entradas, que é dada pela soma dos verdadeiros positivos T_P, T_N, F_P e F_N. Além disso, o F1-Score, que combina sensibilidade (*recall*) e precisão (taxa de predições positivas corretas), é usado como uma métrica de desempenho chave para o UETADS. Essas métricas mencionadas são definidas nas seguints equações:

$$Especificidade = \frac{T_N}{T_N + F_P} \tag{1}$$

$$Sensibilidade = \frac{T_P}{T_P + F_N} \tag{2}$$

$$Acurácia = \frac{T_P + T_N}{T_P + T_N + F_P + F_N} \tag{3}$$

$$Precisão = \frac{T_P}{T_P + F_P} \tag{4}$$

$$F1 - Score = \frac{2 \cdot Precisão \cdot Sensibilidade}{Precisão + Sensibilidade} \tag{5}$$

Para um sistema de detecção de ataques, manter o equilíbrio entre falsos positivos e negativos é fundamental, tornando Especificidade e Sensibilidade métricas chave de desempenho. O F1-score, que oferece uma média harmônica, é benéfico em situações com desequilíbrio de dados entre classes e é amplamente usado em aprendizado de máquina. A acurácia, indicando a proporção de predições corretas do total, é outra métrica de desempenho comumente usada. Coletivamente, essas métricas oferecem uma avaliação da eficácia do sistema em identificar precisamente ameaças e reduzir alertas falsos.

Como *kernel* do OCSVM são usados: Linear, Polinomial, RBF e Sigmoidal. Além disso, são ajustados os hiperparâmetros para identificar a combinação ótima usando a estratégia de busca em grade. Esta abordagem define um intervalo para cada hiperparâmetro e, em seguida, itera através de várias combinações em treinamentos sucessivos [Anyanwu et al., 2022, Budiman, 2019].

Para o treinamento, o modelo usa os primeiros 40% dos dados do conjunto gerado, com o restante servindo como conjunto de teste. Isso equivale a treinar o modelo com dados dos primeiros quatro minutos da conexão do cliente na rede, permitindo que o UETADS seja adaptado a diferentes redes Wi-Fi. No entanto, um aumento no volume de dados de treinamento pode comprometer o desempenho do sistema, pois a exposição prolongada durante a coleta pode interpretar erroneamente as leituras de ataque como comportamento regular da rede.

Na estratégia de busca em grade, os hiperparâmetros ótimos para cada *kernel* são determinados, conforme mostra a Tabela 3. O hiperparâmetro ν, que modula a rigidez da margem do modelo, é crucial na definição da proporção de anomalias potenciais durante o treinamento. Um aumento no ν aproxima a fronteira de decisão do centro de densidade dos dados, diminuindo assim os falsos negativos, mas potencialmente aumentando a taxa de falsos positivos.

Kernel	ν	γ	Grau polinômio	Coeficiente
Linear	0,05	-	-	-
Polinomial	0,05	scale	5	-10,0
Sigmoid	0,05	scale	-	-4,0
RBF	0,0003	auto	-	-

Tabela 3. Hiperparâmetros por *kernel* com maior acurácia.

A Figura 3 mostra as fronteiras de decisão distintas para cada *kernel* usado. A técnica t-NSE, inicializada com Análise de Componentes Principais (PCA), possibilita esta representação gráfica ao reduzir a dimensionalidade dos dados. Apesar de testada durante os treinamentos, este método afeta negativamente a precisão e, assim como o PCA, não é implementado no pré-processamento de dados do modelo final. A Figura 3, embora seja uma projeção, esclarece o desempenho semelhante entre os *kernels* Linear, Polinomial e Sigmoid. Neste cenário, muitas anomalias são facilmente separáveis linearmente das outras entradas, com erros ocorrendo principalmente perto das fronteiras de decisão. O *kernel* RBF, exibindo uma fronteira mais ambígua nesta visualização, identifica efetivamente ataques com características semelhantes às redes legítimas.

O *kernel* RBF se destaca em termos de sensibilidade em comparação com os outros *kernels*, como mostra a Tabela 4. A análise dos falsos negativos revela que os *kernels*

Linear, Polinomial e Sigmoid têm dificuldades para identificar ataques quando os cinco atributos de entrada binários permanecem inalterados. Em termos de especificidade, o RBF tem uma ligeira vantagem, com o Linear em posição próxima. Os erros do RBF ocorrem principalmente devido ao aumento do RTT nos casos de teste, com alguns falsos positivos apresentando RTT fora da faixa inicialmente observada na rede.

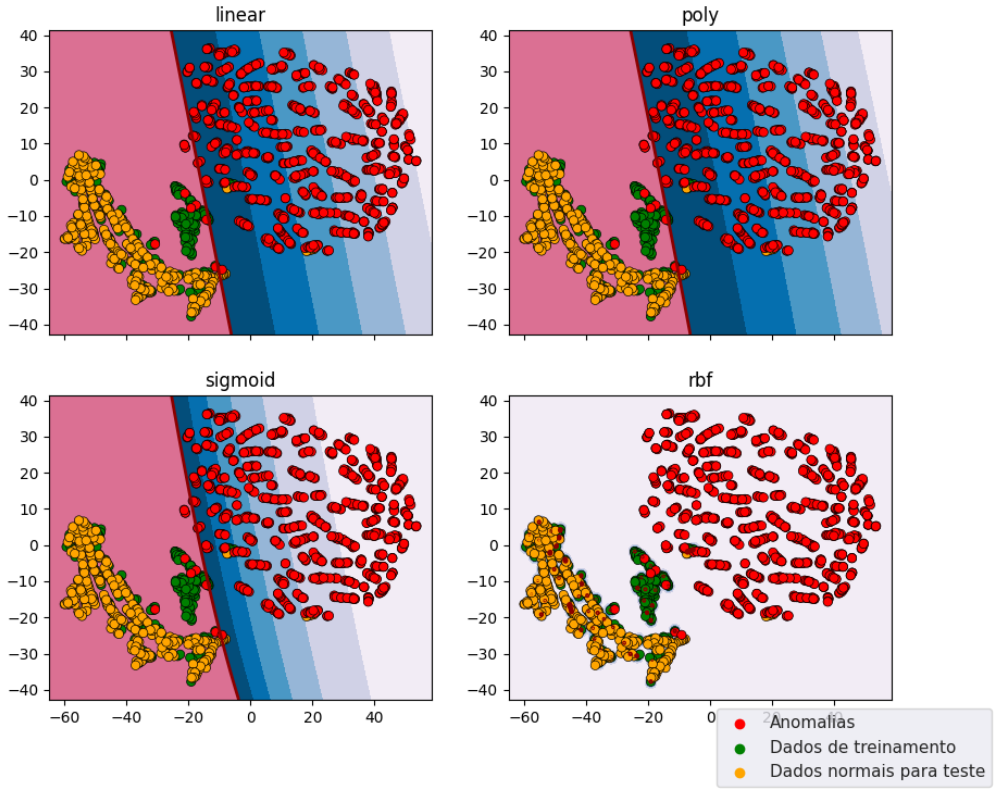

Figura 3. Fronteiras do OCSVM por *kernel*.

Tabela 4. Desempenho por *kernel*.

	kernel			
Métrica	Linear	Polinomial	Sigmoid	RBF
Sensibilidade	85,30%	89,30%	89,40%	**99,50%**
Especificidade	95,84%	94,65%	94,95%	**96,43%**
Acurácia	87,95%	90,65%	90,80%	**98,72%**
F1 score	80,04%	83,61%	83,87%	**97,45%**

Um novo treinamento, usando os mesmos hiperparâmetros, mas com a dimensão de entrada reduzida com base na Tabela 2, mostra que a omissão do atributo menos correlacionado, VAR_RSSI, aumenta a acurácia e o F1-score dos *kernels* Linear, Polinomial e Sigmoid devido à sensibilidade aprimorada. No entanto, essa redução afetou adversamente a especificidade deles, conforme mostra a Tabela 5. Apesar disso, o RBF permanece superior em todas as métricas de desempenho, embora com uma leve diminuição na eficácia em comparação com sua configuração inicial com todos os atributos de entrada.

Tabela 5. Desempenho por *kernel* com redução de dimensão da entrada.

	kernel			
Métrica	Linear	Polinomial	Sigmoid	RBF
Sensibilidade	94,00%	94,70%	94,70%	**99,50%**
Especificidade	87,83%	87,53%	92,28%	**94,36%**
Acurácia	92,44%	92,89%	94,09%	**98,20%**
F1 score	85,42%	86,13%	88,73%	**96,36%**

7. Conclusão e Trabalhos Futuros

Este trabalho introduziu um Sistema de Detecção de Ataques *Evil Twin* Baseado no Usuário, empregando a Máquina de Vetores de Suporte de Classe Única (OCSVM) para detecção de anomalias. Diferentemente dos métodos tradicionais que se concentram na varredura de canais ou no monitoramento centralizado de rede, o sistema proposto se concentra no cliente que o executa, analisando atributos específicos da associação do cliente com o ponto de acesso para detectar possíveis ataques.

O sistema proposto foi implementado em Python 3 e sua eficácia foi avaliada experimentalmente em um ambiente de rede real. Durante um intervalo de tempo especificado, o sistema proposto coleta dados para determinar o comportamento padrão da rede. A precisão do sistema alcança uma taxa de acurácia de 98,72% usando o *kernel* RBF. A sensibilidade do sistema ficou próxima a 90%, mesmo quando apenas os atributos RTT e RSSI foram alterados, demonstrando sua capacidade de identificar ataques com configurações que espelham a rede legítima.

Trabalhos futuros irão se concentrar em transformar o sistema proposto em um sistema que não apenas detecta, mas que também mitiga ataques.Além disso, a variação de parâmetros como o tempo de coleta será avaliada. O sistema proposto também será comparado com outros sistemas encontrados na literatura e que disponibilizem seus códigos-fonte para reprodução dos experimentos.

Agradecimentos

Este trabalho foi realizado com recursos do CNPq, CEFET/RJ, CAPES, FAPERJ, e PGC/UFF.

Referências

Aircrack-ng (2021). Deauthentication. Disponível em `https://www.aircrack-ng.org/doku.php?id=deauthentication`. Acessado em agosto de 2023.

Anyanwu, G. O., Nwakanma, C. I., Lee, J.-M. e Kim, D.-S. (2022). Optimization of RBF-SVM kernel using grid search algorithm for DDoS attack detection in SDN-based VANET. *IEEE Internet of Things Journal*, 10(10):8477–8490.

Apple Inc. Sobre roaming sem fio para empresas. Disponível em `https://support.apple.com/pt-br/HT203068`. Acessado em agosto de 2023.

Budiman, F. (2019). SVM-RBF parameters testing optimization using cross validation and grid search to improve multiclass classification. *Scientific Visualization*, 11(1):80–90.

Cerqueira Júnior, R. L. (2023). Evil twin IDS with One-Class SVM. Disponível em https://gitlab.com/ricardolcj/EvilTwinIDSwithOneClassSVM. Acessado em setembro de 2023.

Faria, V. S., Gonçalves, J. A., Silva, C. A. M. d., Vieira, G. B. e Mascarenhas, D. M. (2020). SDToW: A slowloris detecting tool for WMNs. *Information*, 11(12):544.

Floeter, R. (2004). Hostapd. Disponível em https://man.openbsd.org/hostapd.8. Acessado em agosto de 2023.

Hsu, F.-H., Wu, M.-H., Hwang, Y.-L., Lee, C.-H., Wang, C.-S. e Chang, T.-C. (2022). WPFD: Active user-side detection of evil twins. *Applied Sciences*, 12(16):8088.

Kelley, S. (2022). DNSMasq. Disponível em https://thekelleys.org.uk/dnsmasq/doc.html. Acessado em agosto 2023.

Kitisriworapan, S., Jansang, A. e Phonphoem, A. (2020). Client-side rogue access-point detection using a simple walking strategy and round-trip time analysis. *EURASIP Journal on Wireless Communications and Networking*, 2020(1):1–24.

Li, W. e Liu, Z. (2011). A method of SVM with normalization in intrusion detection. *Procedia Environmental Sciences*, 11:256–262.

Mahfouz, A. M., Abuhussein, A., Venugopal, D. e Shiva, S. G. (2021). Network intrusion detection model using one-class support vector machine. Em *Advances in Machine Learning and Computational Intelligence: Proceedings of ICMLCI 2019*, p. 79–86.

Muthalagu, R. e Sanjay, S. (2021). Evil twin attack mitigation techniques in 802.11 networks. *International Journal of Advanced Computer Science and Applications*, 12(6):38–41.

Nakhila, O., Dondyk, E., Amjad, M. F. e Zou, C. (2015). User-side Wi-Fi evil twin attack detection using SSL/TCP protocols. Em *IEEE Consumer Communications and Networking Conference (CCNC)*, p. 239–244.

Swetha, A. e Shailaja, K. (2020). An effective approach for security attacks based on machine learning algorithms. Em *Advances in Computational Intelligence and Informatics: Proceedings of ICACII 2019*, p. 293–299.

Tian, Y., Wang, S. e Zhang, L. (2021). Convolutional neural network based evil twin attack detection in WiFi networks. Em *MATEC Web of Conferences*, volume 336, p. 08006.

Wang, C., Sun, Y., Lv, S., Wang, C., Liu, H. e Wang, B. (2023). Intrusion detection system based on one-class support vector machine and gaussian mixture model. *Electronics*, 12(4):930.

Wi-Fi Alliance (2020). Wi-Fi optimized connectivity specification v2.0. Relatório técnico, Wi-Fi Alliance.

Yang, K., Kpotufe, S. e Feamster, N. (2021). An efficient one-class SVM for anomaly detection in the Internet of Things. *arXiv preprint arXiv:2104.11146*.