Fortalecendo a Segurança de Redes: Um Olhar Profundo na Detecção de Intrusões com CNN Baseada em Imagens e Aprendizado por Transferência

Pedro Horchulhack[1], **Eduardo Kugler Viegas**[1], **Altair Olivo Santin**[1], **João André Simioni**[1]

[1]Programa de Pós-Graduação em Informática (PPGIa)
Pontifícia Universidade Católica do Paraná (PUCPR)
80.215-901 – Curitiba – PR

{pedro.horchulhack, eduardo.viegas, santin, joao.asimioni}@ppgia.pucpr.br

Abstract. *The application of machine learning (ML) to real-world network intrusion detection has been limited, despite its success reported in the literature. To address the challenges of model updating, this paper presents a new approach that uses convolutional neural networks (CNNs) and transfer learning. To extend the lifetime of the model, the CNN uses flow-based feature expansion. The training data and computational cost are significantly reduced by periodically updating the model using transfer learning. Experiments on 2.6 TB of real-world network traffic demonstrate the feasibility of our proposal. Our proposal improves the average F1 by up to* 0.19 *without updates thereby improving the accuracy of the system.*

Resumo. *A aplicação do aprendizado de máquina (ML) à detecção de intrusão de rede no mundo real tem sido limitada, apesar de seu sucesso relatado na literatura. Para enfrentar os desafios da atualização do modelo, este artigo apresenta uma nova abordagem que usa redes neurais convolucionais (CNNs) e transferência de aprendizagem. A CNN usa uma expansão de características baseada em fluxo para prolongar a vida útil do modelo. Os dados de treinamento e o custo computacional são reduzidos significativamente com a atualização periódica do modelo usando a transferência de aprendizagem. Experimentos com 2,6 TB de tráfego de rede do mundo real demonstram a viabilidade de nossa proposta. Nossa proposta melhora o F1 médio em até* $0,19$ *sem atualização melhorando assim a precisão do sistema.*

1. Introdução

Nos últimos anos, tem havido um crescimento substancial na incidência de ataques de rede. Por exemplo, um relatório do ano de 2023 [Zayo 2023] destacou um aumento de até 150% nos ataques de DDoS nos últimos três anos. Na tentativa de identificar tais ataques, os administradores de rede recorrem a Sistemas de Detecção de Intrusão de Rede (Network Intrusion Detection System (NIDS)) [Bulle et al. 2020], que operam com duas técnicas distintas: *baseadas em assinatura* ou em *baseadas em comportamento* [Sommer and Paxson 2010]. Os NIDS baseados em assinatura procuram padrões bem definidos de ataques nos dados de entrada, identificando apenas eventos previamente catalogados em uma base como a CVE. Por outro lado, a abordagem baseada em comportamento identifica ameaças analisando desvios estatísticos, permitindo a detecção

de novos ataques que se comportam de forma semelhante ao que foi modelado previamente [Gates and Taylor 2006].

As técnicas de detecção de intrusão baseadas em comportamento têm sido extensivamente abordadas na literatura [Sommer and Paxson 2010], tipicamente fazendo uso de métodos de reconhecimento de padrões fundamentados em Aprendizado de Máquina (Aprendizagem de Máquina (AM)) [Molina-Coronado et al. 2020]. Para alcançar o objetivo de detecção, os pesquisadores desenvolvem modelos de AM que se fundamentam no comportamento observado no ambiente, utilizando conjuntos de dados disponíveis para treinamento. Dessa maneira, um modelo de AM treinado com dados reais torna-se capaz de operar efetivamente em ambientes de produção para a detecção de intrusões [Sommer and Paxson 2010].

Na prática, os ambientes de rede apresentam uma série de desafios para as técnicas baseadas em Aprendizado de Máquina (AM), em comparação com áreas em que essas técnicas já foram amplamente exploradas na literatura [Viegas et al. 2019]. Uma das principais dificuldades é a constante variação do tráfego de rede ao longo do tempo, seja devido ao surgimento de novos tipos de ataques ou à criação de novos serviços [Sommer and Paxson 2010]. Essa variação demanda uma atualização frequente do modelo de AM, pois é necessário treiná-lo com conjuntos de dados mais recentes, o que por sua vez implica em um maior custo computacional para processamento dos dados. Como resultado, a disponibilização de um modelo de AM atualizado pode levar desde dias e até semanas. Ademais, os modelos de AM treinados nessas condições podem não conseguir capturar efetivamente a dinamicidade e variabilidade do ambiente, especialmente quando muitas abordagens se baseiam em classificadores superficiais [Molina-Coronado et al. 2020]. Embora esses classificadores de AM possam exibir altas taxas de acurácia durante os testes, eles podem não ser eficientes em representar toda a complexidade do tráfego de rede [Sommer and Paxson 2010].

Nos últimos anos, uma quantidade substancial de aplicações de Aprendizado de Máquina (AM) tem se baseado em Redes Neurais Profundas (Deep Neural Networks (DNNs)), especialmente em Redes Neurais Convolucionais (Convolutional Neural Networks (CNNs)), que têm demonstrado resultados satisfatórios no reconhecimento de imagens e detecção de objetos [Wu et al. 2020]. No entanto, para que as técnicas baseadas em CNNs sejam verdadeiramente eficazes, é necessário um conjunto considerável de características e exemplos nos dados de entrada, o que representa um desafio significativo para sua aplicação. Ao aplicar uma CNN no contexto do tráfego de rede, não há uma quantidade suficiente de características disponíveis para treinamento nos diferentes domínios de interesse. As características comumente derivam de uma janela de tempo, resultando em quantidades que não ultrapassam algumas dezenas. Por outro lado, o processo de treinamento e atualização de modelos de Sistemas de Detecção de Intrusão baseados na Rede (NIDS) requer uma quantidade mínima de dados, principalmente devido à necessidade de tráfego de rede devidamente rotulado, uma tarefa que frequentemente depende de intervenção humana [Viegas et al. 2019].

Em face a este desafio, este trabalho propõe uma nova abordagem para detecção de intrusão de rede, que utiliza Redes Neurais Convolucionais (CNNs) adaptadas para processar fluxos de rede como imagens, além de empregar transferência de aprendizagem durante as atualizações do modelo. O objetivo é estender a vida útil do modelo, ao mesmo

tempo em que é reduzido o custo das atualizações. A implementação ocorre em duas etapas distintas. Na primeira etapa, uma CNN é desenvolvida para expandir as características dos fluxos de rede em um espaço hiperdimensional. Isso é alcançado por meio dos pesos de uma camada oculta de uma rede neural, resultando em uma representação estendida que contribui positivamente para a longevidade do modelo, tanto em termos de custo computacional quanto de acurácia. Na segunda etapa, o modelo da CNN é atualizado periodicamente utilizando o método de transferência de aprendizagem. Nesse processo, o modelo desatualizado é aproveitado, permitindo que as atualizações sejam realizadas com menos recursos computacionais e utilizando uma quantidade menor de dados de treinamento. A ideia principal é aproveitar o conhecimento já armazenado no modelo da CNN anterior para impulsionar o processo de atualização.

2. Fundamentação Teórica

Na administração de redes, os operadores frequentemente contam com o auxílio de ferramentas de Sistemas de Detecção de Intrusão baseados na Rede (NIDS) para identificar ataques de rede [Molina-Coronado et al. 2020]. De maneira geral, as abordagens baseadas em comportamento são construídas em quatro etapas sequenciais: Aquisição de Dados, Extração de Características, Classificação e Alerta. O primeiro módulo é responsável pela coleta de eventos de rede, como pacotes em uma placa de rede (Network Interface Card (NIC)). O segundo módulo foca na extração de informações úteis dos pacotes coletados, formando um vetor de características associado a cada pacote. Por exemplo, o comportamento dos eventos de rede é representado por fluxos de rede que resumem a comunicação entre hosts e serviços dentro de uma janela de tempo específica, como por exemplo, 15 segundos, utilizando características como a quantidade de pacotes trocados entre os hosts. O terceiro módulo utiliza as características coletadas do módulo anterior (o vetor de características) para determinar se os valores contidos representam um evento normal ou um ataque. Nesta etapa, também é possível empregar modelos de Aprendizado de Máquina (AM). Por fim, se um evento é classificado como um ataque, o módulo de Alerta o reporta ao administrador de rede.

Nas últimas décadas, várias abordagens foram propostas para realizar a classificação de eventos, com foco principal na aplicação de algoritmos de Aprendizado de Máquina (AM) [Santos et al. 2023]. Nessas abordagens, um modelo é construído utilizando uma base de dados para treinamento, que idealmente contém várias amostras representativas do comportamento do ambiente real de produção. Na prática, os ambientes de rede exibem um comportamento dinâmico, o que torna a interpretação dos dados coletados uma tarefa complexa. Embora os modelos de AM sejam capazes de reconhecer padrões em grandes quantidades de dados, o treinamento de tais modelos requer uma quantidade significativa de dados rotulados [Sommer and Paxson 2010]. No entanto, mesmo que o modelo de AM seja capaz de lidar com essas complexidades, é necessário atualizá-lo regularmente devido às mudanças no comportamento do tráfego de rede ao longo do tempo [Viegas et al. 2019].

3. Trabalhos Relacionados

A maioria das abordagens de Sistemas de Detecção de Intrusão baseados em Redes Neurais (NIDS) da literatura prioriza principalmente a busca por maior precisão na detecção

de intrusões [dos Santos et al. 2023]. Apesar dos resultados promissores reportados, essas abordagens frequentemente negligenciam os desafios associados às atualizações do modelo devido ao novo tráfego de rede [Viegas et al. 2019] e à potencial influência da variação no tráfego de rede em seus esquemas propostos. Por exemplo, A. E. Kamali *et al.* [Kamali et al. 2023] propuseram a aplicação de uma abordagem que combina Redes Neurais Convolucionais (CNNs) com Redes Neurais Recorrentes (Recurrent Neural Networks (RNNs)) para a detecção de intrusões. Seu esquema resultou em altas precisões de detecção em conjuntos de dados estáticos e desatualizados, porém, em grande parte, ignorou a dinâmica do comportamento do tráfego de rede.

Da mesma forma, L. Yang e A. Shami [Yang and Shami 2022] propuseram o uso de um *ensemble* de CNNs para a detecção de intrusões, resultando em uma melhoria na precisão de detecção em comparação com técnicas tradicionais. No entanto, sua abordagem não considerou as atualizações do modelo. Além disso, para aumentar a confiabilidade da classificação, os autores recorreram ao ajuste fino dos hiperparâmetros do modelo [Viegas et al. 2018]. Um exemplo disso é o trabalho de A. N. Calugar *et al.* [Calugar et al. 2022], em que os autores otimizam os hiperparâmetros da CNN para aumentar a acurácia de classificação em conjuntos de dados estáticos. No entanto, sua abordagem também não considerou variações no comportamento do tráfego de rede e como as atualizações do modelo podem ser realizadas para lidar com essas variações.

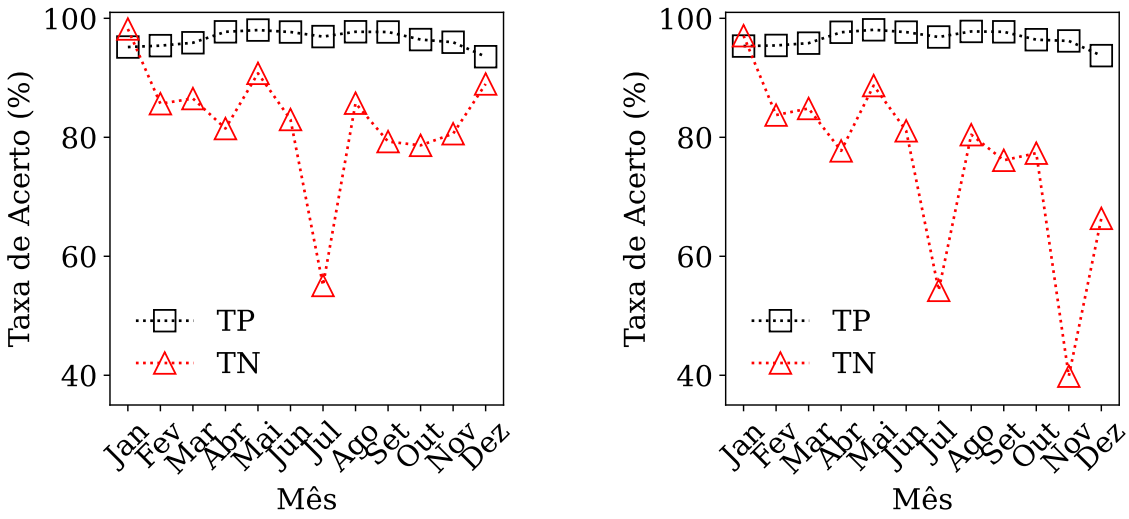
Nos últimos anos, diversos estudos têm questionado a aplicabilidade dos resultados relatados na detecção de intrusões [Horchulhack et al. 2024]. Por exemplo, G. C. Bertoli *et al.* [de Carvalho Bertoli et al. 2023] buscaram uma melhor generalização do modelo, apesar da acurácia. Os autores demonstraram que CNNs podem melhorar a generalização na detecção em vários conjuntos de dados de intrusões. No entanto, o trabalho não abordou como as atualizações do modelo podem ser realizadas. Por outro lado, O. D. Okey *et al.*[Okey et al. 2023] procuraram uma melhor generalização do modelo por meio da transferência de aprendizado usando CNN. Os autores mostraram que modelos pré-treinados podem melhorar a precisão, mas não houve discussão sobre como isso pode facilitar as atualizações do modelo. Uma abordagem semelhante foi introduzida por Sk. T. Mehedi *et al.*[Mehedi et al. 2023], que empregaram a transferência de aprendizado para aprimorar a precisão de detecção em um banco de testes heterogêneo. Apesar de melhorar a precisão e generalização da detecção, eles não abordaram as atualizações do modelo.

4. Definição do Problema

Nesta seção, serão abordados de forma mais detalhada os desafios impostos pela mudança do tráfego de rede em ambientes de produção para NIDS baseados em AM tradicionais. Mais especificamente, primeiro será descrito o dataset que foi utilizado, compreendendo um ano de tráfego de rede real. Em seguida, serão avaliados diferentes modelos de AM para detecção de intrusão, considerando suas taxas de acurácia.

4.1. MAWIFlow

Para viabilizar a avaliação dos esquemas de detecção de intrusões no contexto da mudança no comportamento do tráfego de rede, este trabalho utiliza o conjunto de dados MAWIFlow [Viegas et al. 2019]. Este conjunto de dados foi criado utilizando o Samplepoint-F



(a) Random Forest (RF)

(b) Extra Trees (EXTRA)

Figura 1. Tendência da acurácia ao longo de um ano de classificadores comumente usados sem atualizações periódicas do modelo. O classificador é treinado em janeiro e avaliado nos meses subsequentes sem atualizações.

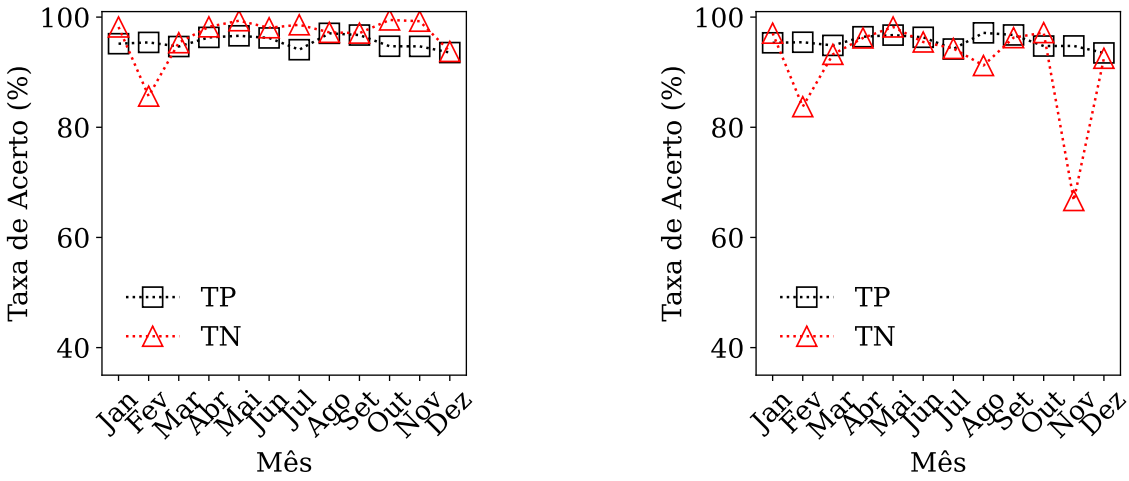
do arquivo MAWI, que consiste em tráfego de rede real coletado diariamente em intervalos de 15 minutos de uma conexão de trânsito entre o Japão e os Estados Unidos. Para os propósitos de avaliação, consideramos todo o tráfego de rede coletado durante o ano de 2014, o qual é posteriormente utilizado para avaliar sistemas de detecção de intrusões amplamente utilizados, baseados em AM.

O conjunto de dados contém mais de 2,6 TB de dados, compreendendo aproximadamente 300 bilhões de pacotes de rede. Para rotular os eventos, foi utilizada uma técnica de AM não supervisionada do MAWILab [Fontugne et al. 2010], que classifica automaticamente registros de entrada como normais ou ataques. Para a tarefa de extração de características, foi utilizado o BigFlow [Viegas et al. 2019], que agrupa eventos em intervalos de 15 segundos e extrai 60 características baseadas em fluxo do conjunto de características Nigel [Williams et al. 2006].

4.2. Lidando com Intrusões de Rede Reais ao Longo do Tempo

Dois classificadores amplamente utilizados foram avaliados: Random Forest (RF) e Extra Tree (ExT). Ambos utilizam uma árvore de decisão como classificador base, implementada considerando o algoritmo C4.5, um fator de confiança de $0,25$ e a métrica de qualidade $gini$ para a separação dos nós. Além disso, ambos utilizaram 100 árvores de decisão, levando em conta os parâmetros citados anteriormente. No processo de treinamento, os dados foram sub-amostrados aleatoriamente, sem repetição, para garantir que a proporção das classes *normal* e *ataque* fossem iguais. Para os classificadores, foram utilizadas suas implementações na versão $v1.3.1$ da API do *scikit-learn*. A avaliação dos modelos foi feita através das suas taxas de Falso Positivo (FP) e Falso Negativo (FN). O FP denota a taxa de eventos *normais* classificados incorretamente como *ataque* e FN denota a taxa de eventos *ataque* que foram incorretamente classificados como *normais*.

A primeira avaliação busca determinar performance dos classificadores mencionados sem que atualizações sejam aplicadas ao longo do tempo. O objetivo da avaliação é



(a) Random Forest (RF)

(b) Extra Trees (EXTRA)

Figura 2. Tendência da acurácia ao longo de um ano com atualizações periódicas do modelo usando classificadores comumente usados, em que o modelo é atualizado mensalmente.

determinar como as mudanças no tráfego da rede podem afetar, ao longo do tempo, a performance dos classificadores. Dessa forma, é possível inferir que tais mudanças afetam, também, técnicas de detecção de intrusão baseadas em AM. Para atingir tal objetivo, os modelos foram treinados usando os dados de Janeiro e, na sequência, foram avaliadas as taxas de FP e FN ao longo do ano. Essa avaliação ajuda a entender o impacto da evolução do comportamento do tráfego da rede na performance de técnicas de detecção de intrusão baseadas em AM.

Por exemplo, a Figura 1 ilustra o comportamento da acurácia dos classificadores escolhidos quando não são realizadas atualizações periódicas. Os resultados mostram que as taxas de erro são significativamente mais baixas em Janeiro, mês que foi utilizado para o treinamento. No entanto, com o passar do tempo, as taxas de erro aumentam gradualmente. Quando observado o classificador RF (Figura 1(a)), em Julho, ele atinge uma taxa de 44% de FP, indicando um crescimento de até $25,4\%$ quando comparado à Janeiro. Esse comportamento se mantém quando observado o classificador ExT, destacando que as atuais abordagens de detecção de intrusão baseadas em AM, como citadas na literatura, não se adaptam adequadamente ao tráfego de rede real.

Por fim, avaliamos a performance dos mesmos classificadores, porém realizando atualizações mensais. Para que isso fosse possível, as atualizações dos classificadores foram feitas no início de cada mês, considerando os dados do mês anterior. Por exemplo, em 1º de Março os modelos são atualizados com os dados de 1º de Fevereiro até o dia 28º de Fevereiro. Essa abordagem permite analisar como as atualizações dos modelos podem melhorar, ao longo do tempo, a performance de técnicas de detecção de intrusão baseadas em AM.

A Figura 2 ilustra o comportamento da acurácia dos classificadores escolhidos quando são realizadas atualizações periódicas mensalmente. Foi observado que há uma melhora na acurácia quando comparados os modelos atualizados com suas versões em que as atualizações não são realizadas (Figura 1 vs. Figura 2). Assim, ao atualizar os modelos

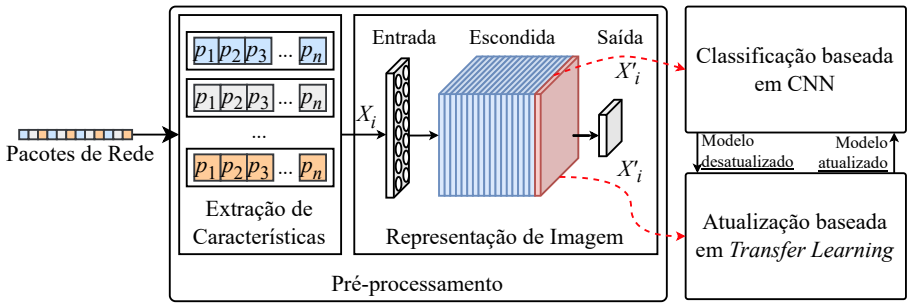


Figura 3. Proposta.

é possível manter as taxas de FP e FN baixas ao longo do tempo. Por exemplo, para o classificador RF com atualizações (Figura 2(a)), foi observada uma taxa de FP de $1,3\%$ em Julho, indicando um decréscimo de $33,5\%$ quando comparada com a sua versão sem atualizações periódicas. Isso indica que tais atualizações melhoram significativamente a performance e confiabilidade de técnicas de detecção de intrusão baseadas em AM.

5. Modelo Proposto

Nossa estratégia para a detecção de intrusões utilizando CNNs, implementada por meio de transferência de aprendizado, tem como ideia-chave que uma CNN pode aprimorar a confiabilidade do sistema por longos períodos, estendendo assim a vida útil do modelo, enquanto a transferência de aprendizado pode facilitar o processo de atualização da CNN. Como resultado, a abordagem proposta pode reduzir o custo computacional associado ao processo de atualização e requerer menos eventos de rede rotulados para o treinamento. Isso é especialmente importante para lidar com a mudança no comportamento do tráfego de rede, pois permite que o modelo se adapte de forma eficiente a essas mudanças sem a necessidade de recriar o modelo do zero a partir de novos dados rotulados. A Figura 3 ilustra o modelo proposto, sendo composta por duas etapas principais: *Classificação Baseada em Imagem* e *Atualização com Transferência de Aprendizagem*.

O módulo *Classificação Baseada em Imagem* é alinhado de acordo com premissas tradicionais de NIDS, em que os eventos da rede são classificados em *normal* ou *ataque*. O procedimento de classificação se inicia com a coleta de pacotes de rede do ambiente monitorado. O comportamento do evento selecionado é obtido por meio de um módulo chamado *Extração de Características*, que extrai características relacionadas ao fluxo. Para aumentar a generalização do modelo, a presente técnica gera um conjunto de características hiperdimensional derivado de uma camada escondida de uma rede neural (Figura 3, X'_i). A hipótese é que a representação estendida do espaço de características contribui para a extensão da vida útil do modelo e da manutenção das acurácias do sistema ao longo do tempo. O conjunto de características resultante é convertido em uma representação em formato de imagem, que serve como entrada para uma CNN para classificação. Dessa forma, a CNN avalia maiores dimensões na entrada, aumentando suas capacidades de generalização, resultando em maior vida útil.

Já o módulo de *Atualização com Transferência de Aprendizagem* é projetado para tratar da constante mudança do comportamento da rede, enquanto diminui os custos relacionados à atualização do modelo. Nesse contexto, a proposta incorpora a atualização do modelo por meio da abordagem de aprendizagem por transferência, onde em ambos

os classificadores desatualizados, a *Rede Neural* e a CNN (Figura 3, *Atualização com Transferência de Aprendizagem*), é aplicado o procedimento durante a atualização dos modelos. A ideia é utilizar os conhecimentos prévios sobre o comportamento da rede do modelo desatualizado, resultando em uma redução substancial no custo computacional.

As subseções a seguir detalharão os procedimentos da proposta do trabalho em relação à classificação e atualização.

5.1. Classificação Baseada em Imagem

As atuais acurácias de classificação do estado da arte são atingidas aplicando arquiteturas de CNNs para as tarefas de detecção de intrusões. Isso se deve à capacidade aprimorada das CNNss de representarem as características do conjunto de dados de treinamento, uma característica valiosa para uma melhor generalização na classificação do tráfego de rede. Diante disso, o modelo proposto utiliza CNNs para realizar a classificação do tráfego de rede. A principal ideia é empregar uma CNN para aumentar a vida útil do sistema com o passar do tempo, mesmo se nenhuma atualização de modelo for realizada. Para alcançar esse objetivo, o módulo proposto de *Classificação Baseada em Imagens* é implementado em duas fases, conforme mostrado na Figura 3.

Ele considera um fluxo de pacotes de rede capturados de uma NIC. O comportamento dos pacotes de rede é extraído, compondo um vetor de características X, associado a um fluxo, de tamanho n (Figura 3, *Extração de Fluxo*). Dado o vetor de fluxo X, em que $X = \{X_1, X_2, ..., X_n\}$, o objetivo é encontrar um rótulo y associado. Para atingir tal objetivo, primeiro é aplicada uma rede neural com uma camada escondida com m neurônios, em que $m > n$ (Figura. 3, *Rede Neural*). Durante o treinamento, a rede neural é otimizada para classificação de fluxos de rede, sendo submetida pelo mesmo procedimento de treinamento da CNN. Assim, as saídas hiperdimensionais não-lineares da camada escondida servem para o processo de classificação em um espaço com mais dimensões. O vetor resultante, denotado por X_i' em que $X_i' = \{X_1, X_2, ..., X_m\}$, é remodelado em uma matriz garantindo, assim, que as características estejam em um formato de imagem. Dessa forma, a matriz é usada como entrada para o modelo CNN, que classifica-a como *normal* ou *ataque*.

O principal benefício da proposta é aproveitar CNN baseadas em imagem para melhorar a confiabilidade e a vida útil do modelo, devido à maior complexidade do modelo. Tal objetivo é atingido por meio de duas etapas. A primeira é o aumento da representação dimensional das características extraindo as saídas da camada escondida de uma *Rede Neural* (Figura 3, X_i'). O vetor resultante melhora a capacidade de generalização da CNN, aumentando, portanto, sua vida útil. Em segundo, é utilizado o vetor de características hiperdimensional, sendo redimensionado para o formato de imagem, como entrada da CNN implantada na proposta. Como resultado, o esquema proposto pode substancialmente aumentar a representação do comportamento dos fluxos de rede antes de usá-lo na etapa de classificação da CNN. Dada a abordagem apresentada, a proposta pode aumentar a generalização e vida útil da CNN.

5.2. Atualização com Transferência de Aprendizagem

O treinamento e a atualização de modelos de CNNs impõem desafios em aplicações NIDS, principalmente pela significativa demanda de dados rotulados, necessitando de frequente intervenção humana.

Para que a atualização do modelo seja viável, existe a demanda pela elaboração de técnicas que tratem desse problema, consumam menos recursos computacionais e gerem dados rotulados para o treinamento.

Como solução, o esquema proposto adota uma estratégia de aprendizagem por transferência para as atualizações, apresentando duas vantagens: (1) O custo computacional é reduzido reutilizando os pesos da CNN desatualizada; (2) A quantidade de dados para treinamento é minimizado, principalmente pelo conhecimento prévio embutido na CNN desatualizada em relação ao comportamento do tráfego da rede.

O modelo proposto no trabalho assume que as atualizações são executadas periodicamente pelo administrador da rede (Figura 3, *Atualização com Transferência de Aprendizagem*). No momento de atualização, a *Rede Neural* é atualizada considerando o modelo antigo e é otimizada dando mais importância à classificação dos novos dados de treinamento. Com a *Rede Neural* atualizada, é gerado um espaço de características hiperdimensional para atualizar a CNN implantada. Neste caso, os pesos desatualizados da CNN são ajustados com base nos novos dados de treinamento gerados pela saída da camada escondida da *Rede Neural*. Por fim, a rede neural e a CNN atualizadas são utilizadas em um ambiente de produção.

6. Avaliação

Nesta seção trataremos do treinamento do modelo e do tratamento de mudanças do tráfego de rede.

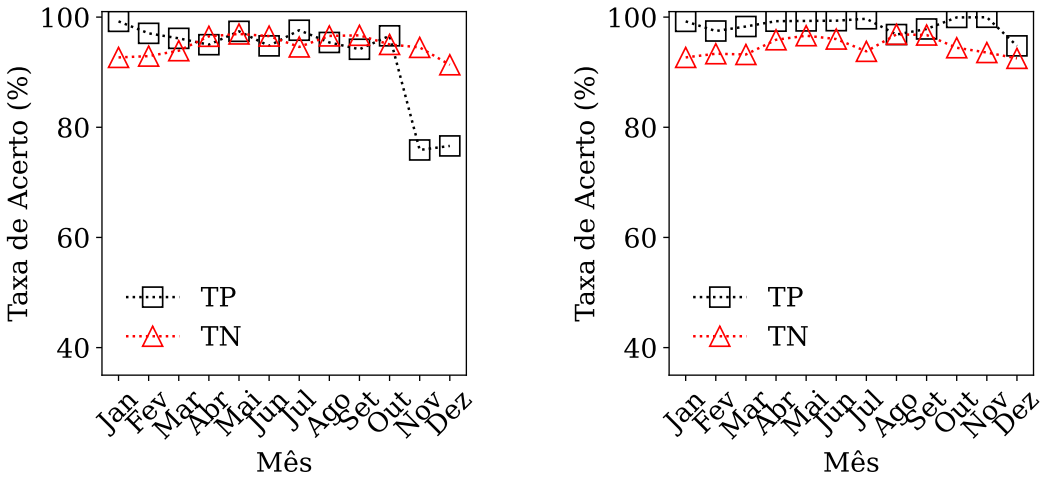
6.1. Treinamento do Modelo

O modelo proposto foi implementado e avaliado usando o dataset anteriormente discutido na Seção 4.1. A *Rede Neural* proposta foi implementada por meio de uma arquitetura de Multilayer Perceptron (MLP), composta por 60 neurônios na camada de entrada, uma camada escondida de 2.048 neurônios e 2 neurônios na camada de saída. O processo de treinamento da rede foi realizada em 1.000 épocas, com a função de ativação $ReLU$ e o otimizador *Adam*. Ainda, a MLP foi implantada usando a API do *scikit-learn* $v1.3.1$

A implementação da CNN foi avaliada a arquitetura GoogLeNet. A entrada da CNN é derivada das saídas da camada escondida da MLP para o processo de treinamento e atualização, como ilustrado na Figura 3 (*Rede Neural*). Neste caso, a camada escondida da MLP é redimensionada de 2.048 para uma matriz $48x48$, usando a função *view* do PyTorch. Além disso, a CNN foi treinada e atualizada usando o otimizador *Adam*, executando sobre 1.000 épocas e com a função *loss categorical cross-entropy* com taxa de aprendizagem de $0,001$. A CNN foi implementada usando a API PyTorch $v1.13.1$.

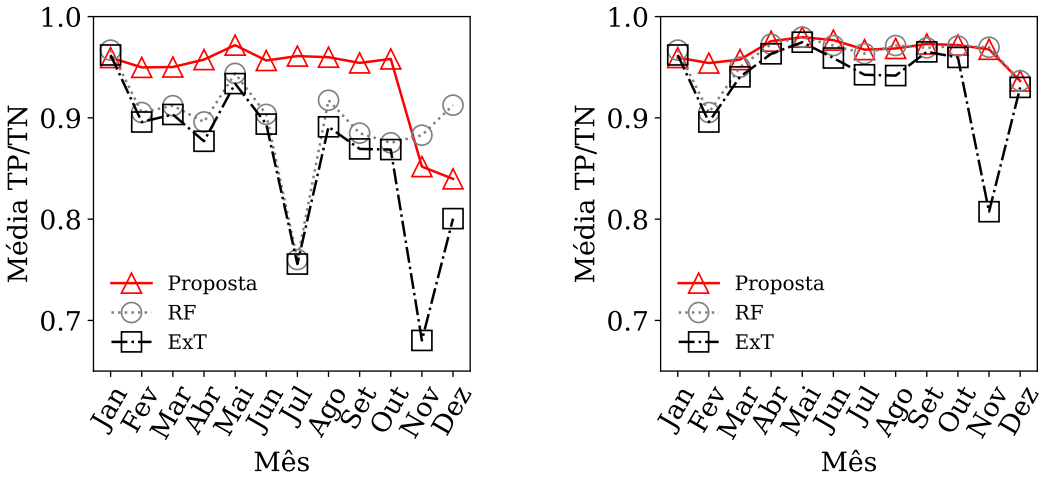
6.2. Tratando Mudanças no Comportamento do Tráfego de Rede

O primeiro experimento busca avaliar a performance da CNN implementada sem que atualizações sejam realizadas. Nesta etapa foi aplicado o mesmo procedimento da Seção 4.2, em que o treinamento do modelo é realizado com os dados de Janeiro e avaliados ao longo do ano sem atualizar a CNN. A Figura 4(a) mostra a acurácia de classificação do modelo proposto neste contexto. É evidente que o esquema proposto mantém suas acurácias de classificação ao longo dos meses, melhorando, significativamente, a vida útil do sistema quando comparado às técnicas tradicionais (Figura 4(a) $vs.$ Figura 1). Por



(a) Sem atualizações (b) Com atualizações

Figura 4. Performance do modelo proposto sobre a base de dados *MAWIFlow*.



(a) Sem atualizações (b) Com atualizações

Figura 5. Comparação de performance sobre a base de dados *MAWIFlow*.

exemplo, o modelo proposto reduz, em média, a taxa de FN em até 27% ao longo do ano avaliado, enquanto a abordagem tradicional, como a RF, reduz, em média, em até 47% sua taxa de FP. Assim, a extração das características hiperdimensionais, usadas como entrada de uma CNN baseada em imagem, pode significativamente aumentar a vida útil do modelo.

O segundo experimento avalia a performance do modelo proposto quando atualizações mensais são realizadas, em que é adotada a estratégia de transferência de aprendizagem na *MLP* e na CNN. Por exemplo, a Figura 4(b) ilustra a acurácia do modelo quando as atualizações são feitas. Assim, é notável que o esquema proposto mantém suas taxas de acurácia consistentes ao longo do ano, quando implementanda uma abordagem de transferência de aprendizagem. O modelo, ainda, apresentou uma média de $5,5\%$ e $1,8\%$ das taxas de FP e FN, respectivamente.

É investigado como o esquema proposto pode melhorar sua confiabilidade em

comparação com as técnicas tradicionais. Em um cenário sem atualizações, a Figura 5(a) mostra o *F1-Score*, a média harmônica entre revocação e precisão, do esquema proposto comparado com as técnicas tradicionais. Em média o modelo proposto atinge um *F1-Score* de $0,93$, apontando uma melhora de $0,18$ e $0,19$ quando comparado aos classificadores RF e ExT, respectivamente, em Junho. Já no contexto em que atualizações são realizadas, a Figura 5(b) mostra o *F1-Score* ao longo do tempo. De forma semelhante, as acurácias da proposta são mais estáveis, mantendo sua eficiência durante o ano avaliado.

7. Conclusão

Técnicas atuais de NIDS baseadas em AM enfrentam dificuldades para modelarem a complexidade do tráfego de rede, resultando em uma perda de performance na taxa de acerto ao longo do tempo, como consequência de alterações no comportamento da rede. O presente trabalho introduziu um novo esquema de NIDS baseado em uma CNN que, de forma realística, captura o tráfego da rede em produção e compreende o seu comportamento, melhorando, assim, a vida útil e a taxa de acurácia do modelo. Ainda, a proposta aproveita a transferência de aprendizagem para facilitar as atualizações do modelo, reduzindo substancialmente o custo computacional associado. Experimentos realizados ao longo de um dataset compreendendo um ano de dados demonstrou a aplicabilidade do modelo. Além disso, a proposta apresenta acurácias maiores do que técnicas tradicionais ao passo que mantém baixo o custo computacional de atualização. Como trabalhos futuros, os autores buscam expandir o modelo proposto para identificar novos padrões na rede de maneira não supervisionada.

Agradecimentos

Este trabalho foi parcialmente financiado pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), processos nº 304990/2021-3 e 407879/2023-4.

Referências

Bulle, B. B., Santin, A. O., Viegas, E. K., and dos Santos, R. R. (2020). A host-based intrusion detection model based on os diversity for scada. In *IECON 2020 The 46th Annual Conference of the IEEE Industrial Electronics Society*. IEEE.

Calugar, A. N., Meng, W., and Zhang, H. (2022). Towards artificial neural network based intrusion detection with enhanced hyperparameter tuning. In *IEEE GLOBECOM*. IEEE.

de Carvalho Bertoli, G., Junior, L. A. P., Saotome, O., and dos Santos, A. L. (2023). Generalizing intrusion detection for heterogeneous networks: A stacked-unsupervised federated learning approach. *Computers & Security*, 127:103106.

dos Santos, R. R., Viegas, E. K., Santin, A. O., and Tedeschi, P. (2023). Federated learning for reliable model updates in network-based intrusion detection. *Computers amp; Security*, 133:103413.

Fontugne, R., Borgnat, P., Abry, P., and Fukuda, K. (2010). MAWILab: Combining diverse anomaly detectors for automated anomaly labeling and performance benchmarking. In *Proc. of the 6th Int. Conf. on emerging Networking EXperiments and Technologies (CoNEXT)*.

Gates, C. and Taylor, C. (2006). Challenging the anomaly detection paradigm: A provocative discussion. In *Proceedings of the 2006 Workshop on New Security Paradigms*, NSPW '06, page 21–29, New York, NY, USA. Association for Computing Machinery.

Horchulhack, P., Viegas, E. K., Santin, A. O., Ramos, F. V., and Tedeschi, P. (2024). Detection of quality of service degradation on multi-tenant containerized services. *Journal of Network and Computer Applications*, 224:103839.

Kamali, A. E., Chougdali, K., and Abdellatif, K. (2023). A new intrusion detection system based on convolutional neural network. In *ICC 2023 - IEEE International Conference on Communications*. IEEE.

Mehedi, S. T., Anwar, A., Rahman, Z., Ahmed, K., and Islam, R. (2023). Dependable intrusion detection system for IoT: A deep transfer learning based approach. *IEEE Transactions on Industrial Informatics*, pages 1006–1017.

Molina-Coronado, B., Mori, U., Mendiburu, A., and Miguel-Alonso, J. (2020). Survey of network intrusion detection methods from the perspective of the knowledge discovery in databases process. *IEEE Transactions on Network and Service Management*, 17(4):2451–2479.

Okey, O. D., Melgarejo, D. C., Saadi, M., Rosa, R. L., Kleinschmidt, J. H., and Rodriguez, D. Z. (2023). Transfer learning approach to IDS on cloud IoT devices using optimized CNN. *IEEE Access*, pages 1023–1038.

Santos, R. R. d., Viegas, E. K., Santin, A. O., and Cogo, V. V. (2023). Reinforcement learning for intrusion detection: More model longness and fewer updates. *IEEE Transactions on Network and Service Management*, 20(2):2040–2055.

Sommer, R. and Paxson, V. (2010). Outside the closed world: On using machine learning for network intrusion detection. In *2010 IEEE Symposium on Security and Privacy*, pages 305–316.

Viegas, E., Santin, A., Abreu, V., and Oliveira, L. S. (2018). Enabling anomaly-based intrusion detection through model generalization. In *2018 IEEE Symposium on Computers and Communications (ISCC)*. IEEE.

Viegas, E., Santin, A., Bessani, A., and Neves, N. (2019). BigFlow: Real-time and reliable anomaly-based intrusion detection for high-speed networks. *Future Generation Computer Systems*, 93:473–485.

Williams, N., Zander, S., and Armitage, G. (2006). A preliminary performance comparison of five machine learning algorithms for practical ip traffic flow classification. *ACM SIGCOMM Computer Communication Review*, 36(5):5–16.

Wu, X., Sahoo, D., and Hoi, S. C. (2020). Recent advances in deep learning for object detection. *Neurocomputing*, 396:39–64.

Yang, L. and Shami, A. (2022). A transfer learning and optimized CNN based intrusion detection system for internet of vehicles. In *ICC 2022 - IEEE International Conference on Communications*. IEEE.

Zayo (2023). The state of ddos attacks ddos insights from q1 & q2, 2023. Technical report, Zayo. Accessed: 2023-10.