Floresta de Decisão Distribuída: Um Sistema de Aprendizado de Máquina Colaborativo Par-a-Par para Detecção de Intrusão em Redes

Lucas Fauster Leite Pereira, Igor Monteiro Moraes e Diogo Menezes Ferrazani Mattos

[1] LabGen/MídiaCom – TET/IC/PPGEET/UFF
Universidade Federal Fluminense (UFF)
Niterói, RJ – Brasil

Abstract. *Distributed machine learning is a solution for collaboratively training models of Intrusion Detection Systems, in which each participant shares only the locally trained model, keeping local data on their devices. This work proposes a Machine Learning System for Distributed Intrusion Detection based on a point-to-point communication topology. The key idea is sharing a Decision Tree model, in which the shared trees make up a Distributed Decision Forest. The work simulates and compares the proposal against a Federated Intrusion Detection System with parameter server communication topology, which deploys a neural network. The simulations show that the Distributed Decision Forest model has a median accuracy of 79% with only one aggregation round. The neural network model reached a median accuracy of 86% but after ten aggregation rounds. The result shows that the Distributed Decision Forest model imposes less processing overhead and greater data privacy to achieve performance comparable to the federated neural network.*

Resumo. *Sistemas de Detecção de Intrusão de nova geração empregam aprendizado de máquina para treinar modelos de forma colaborativa. Os participantes compartilham apenas o modelo treinado localmente, mantendo os dados privados locais nos dispositivos. Este trabalho propõe um Sistema de Aprendizado de Máquina totalmente distribuído para a Detecção de Intrusão, baseado em uma topologia de comunicação par-a-par. A ideia central é o compartilhamento de um modelo de Árvore de Decisão, em que as árvores compartilhadas compõem uma Floresta de Decisão Distribuída. O trabalho simula e compara a proposta com um Sistema de Detecção de Intrusão Federado, com topologia de comunicação de servidor de parâmetros, utilizando como modelo de aprendizado a rede neural. As simulações realizadas mostram que o modelo de Floresta de Decisão Distribuída apresenta a mediana da acurácia em 79% com apenas uma rodada de agregação. O modelo de rede neural atingiu mediana de acurácia de 86%, porém em 10 rodadas de treinamento e agregação. Os resultados mostram que o modelo de Floresta de Decisão Distribuída dispõe de menor sobrecarga de processamento e maior privacidade sobre os dados para alcançar desempenho comparável à rede neural federada.*

1. Introdução

A área da segurança da informação assume importância crescente, pois ataques cibernéticos estão mais frequentes e complexos, causando graves danos. Entre 2020 e

2021, houve aumento médio de 50% dos ataques cibernéticos em todo o mundo, resultando em aproximadamente 293 ataques por semana em 2021[1]. As empresas levam em média 197 dias para detectar uma brecha de segurança e 69 dias para conter uma intrusão aos sistemas afetados[2]. Esse atraso resulta perda de confiança entre parceiros comerciais, baixa de produtividade, multas, além de perdas financeiras. Os sistemas de detecção de intrusão (*Intrusion Detection Systems* - IDS) visam automatizar o processo de monitoramento e análise de ocorrência de eventos em sistemas ou redes de computadores em busca de sinais de intrusão. A maioria dos sistemas de detecção de intrusão realizam análises e filtragem de pacotes de rede pelos cabeçalhos IP e TCP. Contudo, somente filtragem e análise de cabeçalhos não são suficientes, já que os atacantes procuram se ocultar falsificando o IP de origem e alterando dinamicamente parâmetros TCP. Além disso, os sistemas de detecção de intrusão precisam analisar em tempo real uma grande massa de dados que tende a aumentar significativamente com a introdução de dispositivos da Internet das Coisas (*Internet of Things* - IoT). Assim, o uso de algoritmos de aprendizado de máquina é a principal tendência para a evolução dos sistemas de detecção de intrusão.

Aprendizado de máquina é uma das técnicas utilizadas para detectar ataques, pois reconhece padrões relevantes do tráfego de rede e prevê as atividades normais e anormais com base nos padrões aprendidos. Embora modelos de aprendizado de máquina tenham sucesso para IDS, normalmente exigem uma entidade central para processar os dados coletados em toda rede. No entanto, a precisão da previsão diminui quando a escala da rede aumenta, devido à alta taxa de perda de pacotes [Tang et al., 2018]. Por isso, foi proposto o Aprendizado Federado (*Federated Learning* - FL) que consiste no treinamento de um modelo de aprendizado de máquina colaborativamente, preservando a privacidade dos dados [Mothukuri et al., 2021]. O FL permite que os dispositivos aprendam de forma colaborativa, sem a necessidade de compartilhamento de dados com um servidor centralizado, pois evita o compartilhamento dos dados privados [Alazab et al., 2021]. O Aprendizado federado é caracterizado por um grande número de dispositivos com quantidade e distribuição de dados variáveis, dados não independentes e não identicamente distribuídos (*non-IID*) [Neto et al., 2020]. Consequentemente, a introdução do FL para IDS permite que o desenvolvimento de mecanismos de defesa adaptados a diversos dispositivos, com diferentes capacidades de processamento. Contudo, o FL impõe atraso para o aprendizado de novos padrões de ameaças e implica a resolução de um problema distribuído de otimização de parâmetros [Neto et al., 2022].

Este trabalho propõe o Sistema de Detecção de Intrusão baseado na Floresta de Decisão Distribuída. A proposta é um sistema de aprendizado de máquina totalmente distribuído, com privacidade de dados de treinamento, baseado na topologia de comunicação par-a-par. A proposta utiliza o modelo compartilhado de Árvore de Decisão, em que as árvores dos participantes são compartilhadas e compõem uma Floresta de Decisão Distribuída. A proposta é comparada com a abordagem de sistema de aprendizado federado com topologia de comunicação de servidor de parâmetros, baseado em uma rede neural. Ambos os sistemas possuem o objetivo de detectar intrusão e os modelos foram treinados a partir de um conjunto de dados de tráfego de uso de uma rede real. A simulação realizada do sistema de aprendizado federado com topologia de comunicação de servidor

[1]Disponível em https://pages.checkpoint.com/cyber-security-report-2022.html.
[2]Disponível em https://www.ibm.com/security/digital-assets/cost-data-breach-report.

de parâmetros demonstra uma mediana de acurácia de 93%. Na simulação da proposta do sistema com topologia de comunicação par-a-par, o modelo obteve uma mediana de acurácia de 80%, mas o tempo para o aprendizado do modelo foi aproximadamente dez vezes menor, quando comparado ao aprendizado federado.

Diferentemente de trabalhos anteriores [Aragão et al., 2022, Sapio et al., 2021, Sanz et al., 2018, Bellet et al., 2018], o presente trabalho apresenta uma aboradagem totalemente distribuída, em que cada cliente pode compartilhar seus modelos de aprendizado com diferentes clientes e, portanto, convergem localmente para uma floresta de decisão generalizada e com alta acurácia.

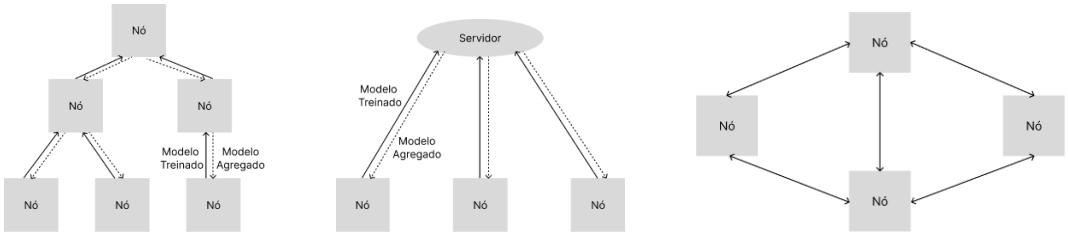
O restante do artigo está organizado da seguinte forma. A Seção 2 apresenta as principais abordagens de aprendizado colaborativo. A Seção 3 propõe a abordagem de aprendizado colaborativo baseado na Floresta de Decisão Distribuída. A implementação do protótipo da Floresta de Decisão Distribuída é discutida na Seção 4. A proposta é avaliada, comparada com a abordagem de Aprendizado Federado e os resultados discutidos na Seção 5. A Seção 6 elenca os trabalhos relacionados. A Seção 7 conclui o artigo.

2. Aprendizado de Máquina Colaborativo

A topologia de aprendizado de máquina mais comumente emprega é o aprendizado centralizado, em que há a necessidade de armazenar todos os dados de treinamento e teste em uma base de dados centralizada. Tal procedimento é bastante custoso tendo em vista que aplicações reais necessitam de grandes massas de dados, superando o armazenamento de dados da ordem de *terabytes*, para se obter alta acurácia e viés limitado [Costa et al., 2012]. Além disso, em muitos casos, os dados são sensíveis e pessoais, o que pode limitar as permissões de armazenamento devidos às exigências da Lei Geral de Proteção de Dados Pessoais (LGPD) [Truong et al., 2021, Tuler De Oliveira et al., 2022]. A LGPD regula as atividades de tratamento de dados pessoais e qualquer atividade que utiliza um dado pessoal na coleta, produção, classificação, acesso, transmissão, processamento, armazenamento, eliminação ou avaliação deve ter o consenso explícito do usuário.

O aprendizado colaborativo visa mesclar conceitos de sistemas distribuídos, redes e aprendizado de máquina. A principal diferença entre o aprendizado de máquina convencional e o aprendizado distribuído é que ao invés dos dados utilizados no treinamento do modelo de máquina serem armazenados em uma única base de dados, os dados permanecem nos dispositivos onde são gerados e apenas o modelo de aprendizado resultante do treino local é compartilhado. Os modelos resultantes são agregados e compartilhados novamente, preservando a privacidade dos dados.

Um sistema de aprendizado de máquina colaborativo pode empregar diferentes abordagens para treinamento e agregação dos modelos [Neto et al., 2020]. Uma abordagem estritamente hierárquica para agregação consiste no envio de modelos para um único nó da rede, onde os modelos são agregados e compartilhados a partir desse nó. Esses sistemas são chamados de *federados* [Neto et al., 2022]. Além disso, há sistemas que permitem a agregação de modelos por um nó intermediário da rede, a partir de modelos compartilhados por todos os nós da rede como em uma topologia de árvore, ou a partir de um modelo parcial, que é compartilhado entre vários nós intermediários. Esses sistemas são chamados sistemas *descentralizados*. Por fim, há sistemas em que todos os nós são independentes e não possuem uma função definida como nos outros sistemas. Assim,



(a) Árvore. (b) Servidor de Parâmetros. (c) Par-a-par.

Figura 1. O aprendizado colaborativo pode ocorrer em diferentes topologias de comunicação e agregação de modelos. Um modelo federado pode ser alcançado através das topologias em árvore (a) ou baseada em um servidor de parâmetros centralizado (b). O modelo totalmente distribuído é alcançado com uma topologia de agregação par-a-par (c).

todos os nós treinam, agregam e compartilham os modelos entre si. Esses são sistemas *totalmente distribuídos*.

As topologias de comunicação entre os nós de um sistema de aprendizado colaborativo são diversas, variando da organização hierárquica em árvore, federada em um servidor de parâmetros ou totalmente distribuída em uma rede sobrecamada par-a-par. A **topologia em árvore** tem a vantagem de ser fácil de escalar e gerenciar, pois cada nó se comunica apenas com seus nós pai e filhos. Dessa forma, os nós em uma árvore agregam seus modelos locais com os de seus filhos e passam o modelo resultante para o seu pai, para calcular um modelo global. A topologia em árvore pode ser vista na Figura 1(a). A **topologia federada de servidor de parâmetros** (*Parameter Server* - PS) possui um conjunto descentralizado de clientes e um conjunto centralizado de servidores que exercem a função de agregar e atualizar os modelos dos clientes. Todos os parâmetros dos modelos de cada cliente são armazenados em um banco de dados em cada servidor, a partir do qual todos os clientes leem e gravam, em um armazenamento de chave-valor. Uma vantagem é que todos os parâmetros dos modelos estão em uma memória compartilhada global, o que facilita a análise do modelo. Contudo, uma desvantagem é que os servidores de parâmetros são um gargalo na rede, pois precisam lidar com toda agregação de parâmetros e centralizam a comunicação, gerando um ponto único de falha. A **topologia federada de servidor de parâmetros** pode ser considerada um caso especial da **topologia em árvore** com apenas duas camadas de hierarquia. A topologia de servidor de parâmetros é mostrada na Figura 1(b). A **topologia par-a-par** estabelece que cada nó participante do aprendizado tem sua própria cópia dos parâmetros do modelo e se comunica diretamente com seus pares. Isso permite uma maior escalabilidade do que um modelo federado e a eliminação de pontos únicos de falha. Um exemplo de implementação da topologia par-a-par para o aprendizado distribuído é a Aprendizagem por Fofoca (*Gossip Learning* - GL) [Shi et al., 2018], que é baseada na ideia de que os modelos são móveis e realizam caminhos aleatórios independentes através da rede par-a-par. Como isso, os nós executam processamento paralelo de dados e modelos. Os modelos evoluem de forma diferente e precisam ser agregados. No GL, isso acontece continuamente nos nós, combinando o modelo atual com um *cache* limitado de visitantes anteriores. A topologia par-a-par é representada na Figura 1(c).

Um dos principais desafios dos modelos de aprendizado colaborativo é a agregação dos modelos locais em modelos globais. Os principais métodos de agregação

de modelos colaborativos são a média federada e o algoritmo de *bagging*. A média federada foca na agregação de modelos federados de aprendizado, Figuras 1(a) e 1(b), enquanto o algoritmo de *bagging* foca na agregação de classificadores e, portanto, tem um amplo espectro de utilização em topologias par-a-par, Figura 1(c).

A Média Federada (*Federated Averaging* - FedAvg) foi o primeiro algoritmo de agregação de modelos locais para aprendizado de máquina federado [Lim et al., 2020] e é baseada na agregação de vetores de gradiente provenientes do algoritmo de otimização do gradiente descendente estocástico (*Stochastic Gradient Descent* - SGD). O algoritmo FedAvg foi inicialmente implementado no Gboard [Hard et al., 2018], o teclado inteligente da Google, para melhorar o modelo de previsão de próxima palavra. O algoritmo FedAvg se baseia no SGD devido ao avanço das aplicações de aprendizado profundo, em que o método de aprendizado converge na direção do SGD [McMahan et al., 2017]. No algoritmo FedAvg é selecionada uma porção S de participantes em cada rodada de comunicação e calcula-se o gradiente da perda sobre todos os dados mantidos por esses participantes. Caso $S = 1$, isso corresponderá, então, ao gradiente descendente determinístico, já que nesse caso ocorre a seleção de todos os participantes. Cada participante computa $\nabla F_n(w_t)$, que são os gradientes em seus dados locais para o modelo atual w_t, e o servidor agrega esses gradientes aplicando a atualização $w_{t+1} \leftarrow w_t - \eta \sum_{n=1}^{N} \frac{D_n}{D} \nabla F_n(w_t)$. O hiper parâmetro η é a taxa de aprendizado, que influencia diretamente a velocidade de convergência do gradiente, t é a identificação da rodada de atualização, n é a identificação dos clientes participantes, N é a quantidade de clientes participantes, F é a função de perda e D é o peso da média. Um tipo de agregação equivalente e mais utilizado é $\forall n, w_{t+1}^n \leftarrow w_t^n - \eta \nabla F_n(w_t^n)$ e, então, $w_{t+1} \leftarrow \sum_{n=1}^{N} \frac{D_n}{D} w_{t+1}^n$. Cada cliente realiza localmente uma ou mais etapas de treinamento no modelo local, usando seus dados, e o servidor obtém a média ponderada dos modelos resultantes (agregação). A estrutura do algoritmo permite adicionar mais computação para cada cliente, realizando a atualização local várias vezes antes da etapa de agregação. A média federada é o principal algoritmo para a realização do Aprendizado Federado [Neto et al., 2020, Neto et al., 2022].

O algoritmo de *Bagging* treina vários modelos diferentes em diferentes conjuntos de dados de treinamento [Medeiros et al., 2020]. Isso pode ser feito de diversas maneiras, sendo a mais comum utilizar partições aleatórias de um conjunto de dados e aplicar o treinamento de modelos de árvores de decisão sobre esses dados. Uma vez que os modelos tenham sido treinados, pode-se combinar suas previsões através de métodos de votação. Esse algoritmo possui três hiper-parâmetros principais que precisam ser definidos antes do treinamento: o número da parcela de dados K, o número de árvores de decisão T e o número de rótulos (classes) do conjunto de dados. Contudo, o algoritmo de *Bagging* é definido para o uso de agregados de classificadores (*emsemble classifier*) em modelos de aprendizado de máquina centralizado. Este trabalho estende a aplicação do algoritmo de *Bagging* para o cenário distribuído em uma rede par-a-par.

3. Sistema de Detecção de Intrusão baseado em Floresta de Decisão Distribuída

A proposta do Sistema de Detecção de Intrusão baseado em Floresta de Decisão Distribuída recorre a uma topologia de comunicação em rede par-a-par. Assim, os clientes

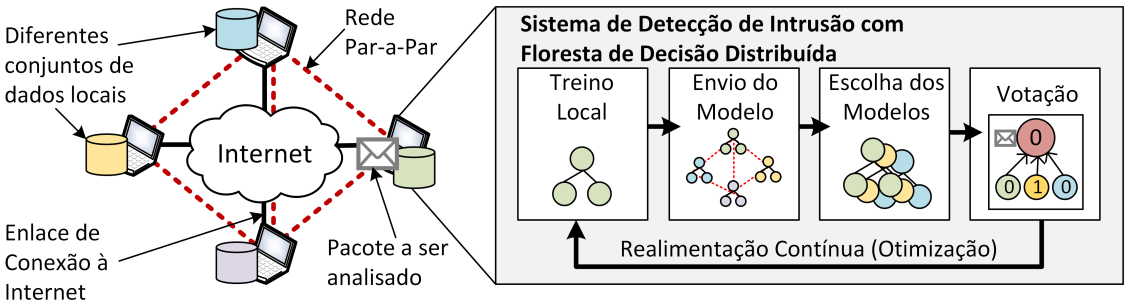


Figura 2. Sistema de Detecção de Intrusão baseado em Floresta de Decisão Distribuída. O nós participantes do sistema de detecção de intrusão estabelecem uma rede sobrecamada par-a-par. Os modelos locais são treinados sobre dados privados e compartilhados na rede par-a-par. Pacotes de rede são analisados pelo modelo composto pelas k melhores árvores e classificados de acordo com a votação das árvores do modelo.

se comunicam entre si diretamente, através de uma rede sobrecamada estabelecida sobre a Internet, e compartilham o modelo de aprendizado de máquina. Nesse sentido, a proposta usa a Árvore de Decisão (DT) como modelo de aprendizado de máquina para detecção de intrusão em rede. O modelo de árvore de decisão apresenta bom desempenho para classificação de tráfego de rede, além de apresentar baixa sobrecarga de processamento e alta capacidade de compreensão [Medeiros et al., 2020]. Após o modelo de cada cliente ser treinado, inicia-se o processo de compartilhamento desses modelos, em que os clientes armazenam os modelos de outros clientes em uma lista, que então será usada para predizer o tipo do fluxo de dados da rede utilizando o método *Bagging*. Essa lista de Árvores de Decisão é chamada, neste trabalho, de Floresta de Decisão Distribuída. A Figura 2 explicita o funcionamento da proposta.

Em cenários de aprendizado centralizado, o modelo é treinado em um ambiente com alto poder computacional, como servidores em nuvem, e *a posteriori* é somente aplicado em dispositivos com processamento restrito. Contudo, no cenário de aprendizado colaborativo, o treinamento e a aplicação do modelo ocorrem em retroalimentação contínua nos dispositivos com recursos restritos. Devido ao fato de o aprendizado colaborativo ser fortemente focado em dispositivos de Internet das Coisas (IoT), a restrição de processamento dos nós deve ser considerada na definição do modelo de aprendizado local a ser adotado. Assim, o modelo de Árvore de Decisão é empregado, pois é um modelo prático para aproximar funções com valores discretos e, em comparação com modelos baseados em rede neural, é um modelo mais simples em termos de cálculos matemáticos e, por consequência, exige menos poder computacional para treiná-lo. O algoritmo de aprendizado desse modelo é robusto para dados com ruídos, valores fora do domínio, ausência de valores, inconsistências, entre outras anomalias dos dados.

O sistema é composto por vários nós participantes, em que cada nó possui um conjunto de dados local. Inicialmente cada nó treina o modelo de Árvore de Decisão sobre o conjunto de dados que possui. Assim que o modelo é treinado, inicia-se o procedimento de compartilhamento dos modelos. Cada nó envia seu modelo para todos os outros nós participantes do treinamento. Em posse dos modelos dos nós vizinhos, cada nó constrói um conjunto de Árvores de Decisão, denominado Floresta de Decisão Distribuída, que contém as k Árvores de Decisão dos nós vizinhos. Em seguida, o nó utiliza sua Floresta de Decisão Distribuída para predizer os fluxos da rede utilizando o algoritmo de *Bagging*,

no qual todas as árvores de decisão da Floresta de Decisão Distribuída predizem individualmente a classe do fluxo. A classe mais recorrente nas predições é a escolhida para rotular o fluxo. O algoritmo de *Bagging* realiza a votação de predições entre todas as árvores que compõem o modelo agregado.

4. Implementação do Protótipo da Proposta

Um protótipo do sistema proposto foi implementado em linguagem Python. As árvores de decisão são instâncias da classe *DecisionTreeClassifier* da biblioteca *sklearn*. Considera-se a profundidade máxima das árvores (*max-depth*) com valor igual 7, com o intuito de evitar, ou diminuir, o sobreajuste (*overfitting*) do modelo com os dados de treino. Valores variando de 5 a 15 foram testados para o parâmetro de profundidade, utilizando o método *GridSearch* da biblioteca *sklearn*. A profundidade 7 foi empiricamente escolhida devido ao melhor compromisso entre acurácia e generalidade dos classificadores. O parâmetro de escolha do método de partição dos nós da árvore (*criterion*) recebe o valor *gini*, referindo-se à função que mede a qualidade da separação de galhos dessa árvore. A função de impureza *gini* foi escolhida por ter gerado melhores resultados em comparação à função alternativa de entropia. Paralelamente, é introduzido um novo atributo, *local-forest*, que representa a Floresta de Decisão Distribuída que o nó mantém. O nó recebe uma lista contendo inicialmente o modelo instanciado por ele e, na sequência, acrescenta os modelos compartilhados pelos outros nós da rede.

Ao inicializar o sistema, os nós da rede iniciam a fase de treinamento do modelo da árvore de decisão com os dados locais. Cada nó pré-processa seus dados, treina o modelo de árvore de decisão com seus dados pré-processados e testa o modelo com as métricas acurácia, precisão, revocação e *f1-score*. Após o treinamento dos modelos, os nós da rede compartilham seus modelos entre si. No protótipo implantado, considera-se que o compartilhamento é entre todos os nós, contudo o compartilhamento pode ser realizado somente com um subconjunto reduzido de nós da rede, de acordo com métricas específicas de afinidade e de custo de compartilhamento[3]. No momento de compartilhamento dos modelos, cada nó adiciona ao seu conjunto de árvores de decisão os modelos que obtiverem as melhores acurácias sobre os dados de teste locais. Esse conjunto de modelos é chamado Floresta de Decisão Distribuída e, empiricamente, foi definido um total de 5 árvores para as florestas de todos os nós da rede. Outras métricas podem ser utilizadas como fatores de decisão para a inserção de árvores nos conjuntos, como por exemplo, as métricas de precisão e revocação. Em seguida, cada nó testa sua Floresta de Decisão Distribuída nos dados do conjunto de dados reservados para teste.

5. Avaliação da Proposta

A avaliação da proposta foca em implementar e analisar dois sistemas de detecção de intrusão em rede baseados em aprendizado de máquina colaborativo. O primeiro é o sistema federado com topologia de comunicação deservidor de parâmetros e o segundo, a proposta de aprendizado distribuído com topologia de comunicação par-a-par. Diferentes modelos de aprendizado de máquina são utilizados juntamente com diferentes métodos de agregação de modelos. Os modelos testados são a rede neural com agregação por média

[3]A definição do conjunto ótimo de nós para compartilhamento dos modelos está fora do escopo deste trabalho.

federada e árvores de decisão com agregação por *bagging*. Para fins de comparação, as métricas acurácia, função de perda da entropia cruzada, precisão, revocação, medida F1 (*F1-score*) e área abaixo da curva de Característica de Operação do Receptor (*Receiver Operating Characteristic* - ROC) são utilizadas.

O trabalho realiza a simulação dos sistemas de aprendizado colaborativo através de um simulador de evento discreto, implementado em linguagem Python. Para tanto, foi utilizada a biblioteca de código aberto TensorFlow[4]. Essa biblioteca fornece uma extensa base de algoritmos de aprendizado de máquina e é uma das mais utilizadas atualmente. Outra biblioteca de aprendizado de máquina, a Scikit-learn[5], foi utilizada por inclui algoritmos de classificação, regressão e agrupamento. A biblioteca Pandas[6] é uma biblioteca para manipulação e análise de dados. A biblioteca é principalmente usada para aprendizado de máquina, pela facilidade que oferece através da estrutura de dados *DataFrame*.

O conjunto de dados utilizado corresponde ao tráfego de acesso de 373 usuários de banda larga fixa de uma grande operadora de telecomunicações na Zona Sul da cidade do Rio de Janeiro, RJ [Lopez et al., 2017]. A base de dados analisada foi criada a partir da captura de pacotes brutos com a ferramenta *tcpdump*, contendo informações reais de tráfego IP (*Internet Protocol*) dos usuários residenciais que foram tratados por um sistema de detecção de intrusão (*Intrusion Detection System - IDS*) de rede, e, posteriormente, resumidos em fluxo de rede utilizando a ferramenta *flowtbag* [7]. A marcação dos dados com o sistema de detecção de intrusão estabelece a verdade básica sobre a qual os algoritmos de aprendizado de máquina extraem conhecimento (*ground truth*).

O conjunto de dados inicialmente possuía um total de 590 classes de tipos de fluxos de ataque e uma classe do tipo fluxo normal. O conjunto de dados foi tratado para marcar os dados com apenas duas classes, a classe 0 referente ao fluxo normal, e a classe 1 correspondente a todos os tipos de ataques de rede registrados durante coleta dos dados. Por se tratar de um conjunto de dados real, há a questão da diversidade dos dados locais dos participantes. A heterogeneidade de dispositivos e a heterogeneidade estatística dos dados geram um conjunto de dados de distribuições estatísticas diversas entre os participantes do sistema. As características em um conjunto de dados *non-IID* (*non Independent and Identicaly Distributed*) são dependentes e não distribuída de forma idêntica. Assim, cada característica tem funções de probabilidade de distribuição distintas e pode haver dependência estatística entre diferentes características de diversos participantes. Esse fato implica o possível enviesamento do aprendizado do modelo global.

As simulações dos sistemas foram realizadas em um computador com processador Intel(R) Core(TM) i7-9700 CPU @ 3.00GHz, 32GB de memória RAM e GPU GeForce GTX 1660 SUPER 6GB. Foram utilizados, no total, 6GB de dados para o treinamento dos modelos, correspondendo a 5 dias de fluxos de dados coletados, com mais de 23 milhões fluxos[8] de dados. As simulações consideram 10, 20 ou 30 clientes do conjunto

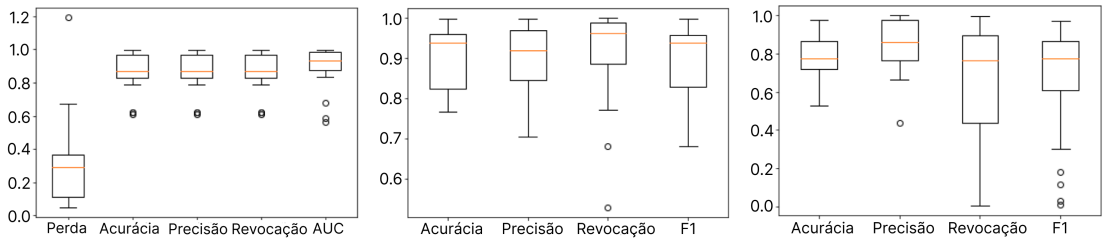
[4]Disponível em https://www.tensorflow.org/.

[5]Disponível em https://scikit-learn.org/stable/index.html.

[6]Disponível em https://pandas.pydata.org/docs/index.html.

[7]Disponível em https://github.com/DanielArndt/flowtbag.

[8]O conjunto de dados analisado considera como fluxo de dados a sequência de pacotes unidirecional identificada pela 5-tupla: enderenço IP de origem, de destino, protocolo de transporte, porta de origem e de destino.



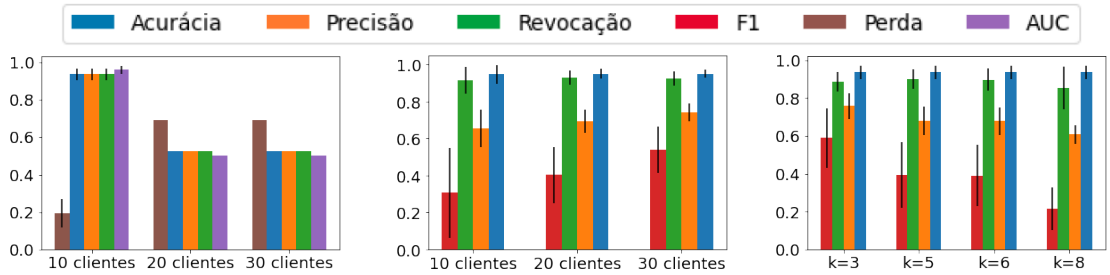
(a) Aprendizado Federado. (b) Árvores de Decisão. (c) Floresta de Decisão Distribuída.

Figura 3. Métricas finais de acurácia, precisão, revocação e *F1-score* do (a) Aprendizado Federado com Rede Neural, das (b) Árvore de Decisão locais do nós e das (c) Florestas de Decisão dos nós participantes do aprendizado de máquina distribuído do sistema com topologia par-a-par.

de dados participando do aprendizado colaborativo. Assim, em cada cenário, seleciona-se um determinado número de clientes aleatoriamente, dentre os 373 clientes monitorados no conjunto de dados, para participarem do aprendizado colaborativo. Cada cliente é representado pelo seu endereço IP e os fluxos direcionados ao participante são aqueles em que a origem ou o destino incluam o IP do cliente participante.

Em um primeiro momento, foi simulado um IDS baseado no aprendizado federado com topologia de comunicação de servidor de parâmetros, para 30 clientes participantes. Esse primeiro resultado estabelece uma linha base de comparação para os resultados do aprendizado distribuído. Após 10 rodadas de treinos locais seguidas de agregações globais, os resultados da simulação do sistema com topologia de servidor de parâmetros mostram que o modelo agregado obteve uma mediana de acurácia, precisão e revocação de 86% e uma mediana de AUC igual a 93%, como mostrado na Figura 3(a). Esses resultados indicam que o modelo agregado obteve um alto desempenho ao predizer os fluxos do conjunto de dados de teste dos clientes, apesar de haver valores discrepantes (*outliers*) no intervalo entre 60% e 70%.

No sistema proposto de detecção de intrusão distribuído com topologia de comunicação par-a-par (Floresta de Decisão Distribuída), cada cliente realiza uma rodada de treinamento do modelo de árvore de decisão local (*Decision Tree* - DT). A simulação considera o cenário de 30 clientes participantes. As métricas locais de cada nó participante são calculadas e a Figura 3(b) apresenta os valores das métricas referentes à predição das árvore de decisão locais dos nós participantes do sistema sobre o conjunto de dados destinado ao teste. O valor da mediana da acurácia desses modelos é igual a 93%, da mediana da precisão é igual a 91%, da mediana da revocação é igual a 96% e da mediana da medida F1 igual a 93%. Na Figura 3(c) são representados os valores para as métricas referente à predição da Floresta de Decisão Distribuída dos nós participantes sobre o conjunto de dados destinado ao teste. A Árvore de Decisão de um nó é um modelo treinado apenas no conjunto de dados desse nó, já a Floresta de Decisão Distribuída é um modelo que corresponde a um conjunto de Árvores de Decisão de vários nós da rede. O valor da mediana da acurácia desses modelos é igual a 79%, da mediana da precisão é igual a 83%, da mediana da revocação é igual a 79% e da mediana da medida F1 igual a 80%. A diferença nos valores das métricas da árvore de decisão e da Floresta de Decisão é devido ao fato dos conjuntos de dados dos nós serem *non-IID*, o que faz com que a predição das árvores de decisão, que foram treinadas com o conjunto de dados dos nós vizinhos, tenham um



(a) Aprendizado Federado com redes neurais em função do número de clientes participantes do treinamento.

(b) Métricas para o Aprendizado Distribuído com Florestas de Decisão Distribuídas em função do número clientes no treinamento.

(c) Aprendizado Distribuído em função do tamanho k da Floresta de Decisão Distribuída.

Figura 4. Metricas de acurácia, precisão, revocação, *F1-score* e *AUC-score* das redes neurais (a) e das Florestas de Decisão Distribuídas (b) em função do número de participantes dos sistemas e (c) pelo número de arvores na Floresta de Decisão Distribuída. O aumento do número de participantes tende a implicar melhora no desempenho da Floresta de Decisão Distribuída.

desempenho menor ao predizer o conjunto de dados de teste de um nó participante. A agregação de todos os modelos em um só não é realizada, como é feito com a Média Federada, implicando menor quantidade de informação do modelo distribuído em relação ao federado. No entanto, utilizar as árvores de decisão treinadas em conjuntos de dados distintos *non-IID* diminui as chances de as predições serem enviesadas, aumentando a confiabilidade das predições feitas pelas Florestas de Decisão Distribuída.

A partir dos resultados mostrados na Figura 3(a) e Figura 3(c) pode-se comparar o desempenho do treinamento dos modelos de aprendizado federado e distribuído. Apesar de o desempenho da rede neural do sistema com topologia de servidor de parâmetros ser maior que o desempenho da Floresta de Decisão Distribuída, ressalta-se que o tempo para que cada modelo fosse treinado para alcançar tais resultados foram bastante distintos. O modelo de Floresta de Decisão Distribuída alcança tais resultados em aproximadamente 99.3 segundos, em comparação ao modelo de rede neural que alcança tais resultados em aproximadamente 952.5 segundos, ambas as simulações realizadas nas mesmas condições e com a mesma quantidade de dados. No entanto, as árvores de decisão necessitam de uma grande quantidade de dados de treinamento, comparadas a outros modelos, para alcançar um desempenho ótimo e com pouco enviesamento [Medeiros et al., 2020]. As redes neurais permitem o treinamento do modelo repetidas vezes sobre o mesmo conjunto de dados, resultando em bom desempenho e com pouco enviesamento, reduzindo assim a quantidade de dados necessários para se alcançar tais resultados[Liu e Lang, 2019].

As Figuras 4(a) e 4(b) comparam métricas das redes neurais e Florestas de Decisão Distribuídas pela quantidade de participantes no treinamento dos modelos do sistema com topologia de comunicação de servidor de parâmetros e sistema com topologia de comunicação par-a-par. Os valores apresentados são as médias, com intervalo de confiança de 95%. Verifica-se que, à medida que a quantidade de participantes aumenta, o desempenho do modelo de rede neural tende a diminuir. Por outro lado, o desempenho do modelo de Floresta de Decisão Distribuída tende a aumentar, o que mostra que a Floreta de Decisão Distribuída, utilizada no sistema com topologia de comunicação par-a-par, é mais escalável em comparação à rede neural utilizada no sistema com topo-

logia de comunicação de servidor de parâmetros. Isso ocorre devido às características do método de agregação utilizado na Floresta de Decisão Distribuída, pois a votação é pouco impactada por modelos enviesados.

A Figura 4(c) mostra a variação nas métricas de desempenho da Floresta de Decisão Distribuída em função do tamanho k da Floresta, ou seja, o número de árvores selecionadas para compor a floresta de decisão local de cada nó. O tamanho da Floresta de Decisão Distribuída foi selecionado a partir da observação do desempenho do modelo para cada tamanho testado. Observa-se que à medida em que o tamanho da floresta aumenta, a média das métricas de precisão e medida F1 diminuem, porém a média da métrica de revocação aumenta até $k = 6$. A diminuição da média das métricas de precisão e medida F1 é explicada pelo gradativo aumento da generalização do modelo em predizer fluxos com distribuição diferente da distribuição dos fluxos em que o modelo foi treinado localmente. Contudo, a generalização permite que o modelo adquira maior revocação ao predizer esses fluxos, ao custo de menor precisão com os dados locais.

6. Trabalhos Relacionados

Sistemas de Detecção de Intrusão baseados em mecanismos de aprendizado de máquina são uma tendência atual. Sanz *et al.* propõem um sistema de detecção, em tempo real, de ameaças distribuídas de rede baseado em aprendizagem enriquecida por grafos [Sanz et al., 2018]. Diferentes métricas são extraídas a partir de uma análise por grafos em janelas de tempo, que são incorporadas às características originais de fluxos durante o pré-processamento. Contudo, a aplicação de grafos nesse sistema se mostrou mais custosa em comparação a modelos de aprendizado de máquina convencionais. Aragão *et al.* avaliam e comparam diferentes modelos de aprendizado de máquina aplicados à identificação de ameaças a dispositivos *IoT* [Aragão et al., 2022]. A análise se baseia no treinamento e otimização dos modelos, além de técnicas de pré-processamento de dados, tais como algoritmos para padronização de rótulos das classes, a codificação de características categóricas (*one-hot encoding*) e redução de dimensão através da análise de componentes principais No entanto, em um cenário de aprendizado de máquina distribuído, essas técnicas de pré-processamento não são aplicáveis, pois a heterogeneidade do conjunto de dados dos participantes aumenta e impacta a acurácia do modelo. Liu *et al.* utilizam o algoritmo *K-Means* juntamente com a Floresta Aleatória (*Random Forest*) para solucionar o problema de detecção de intrusão em cascada [Liu et al., 2021]. Os autores comparam a proposta com algoritmos de aprendizado profundo (*Deep Learning*), tais como redes neurais profundas e redes neurais convolucionais. Khan *et al.* propõem um método de detecção de ataques baseado em *Spark ML* e em uma rede neural convolucional de memória longa de curto prazo (*Long Short Term Memory* - LSTM) [Khan et al., 2019].

Sapio *et al.* desenvolvem o arcabouço SwitchML, que integra outros arcabouços de software que possuem algoritmos de aprendizado de máquina como *PyTorch* e *TensorFlow* [Sapio et al., 2021]. O arcabouço proposto visa acelerar a comunicação e melhorar o treinamento de modelos de redes neurais profundas. Essa implementação usa a agregação de modelos de redes neurais profundas (*Deep Neural Networks* - DNN), com algoritmo do gradiente descendente estocástico (*Stochastic Gradient Descent* – SGD), aplicado separadamente em diferentes porções dos dados de entrada (*mini-batch*), desconsiderando a ordem, sem afetar a precisão do resultado final, de forma paralelizada. Contudo, o modelo desconsidera a ordem dos dados, o que implica perda da relação

temporal entre fluxos. Chamikara *et al.* argumentam que o aprendizado de máquina distribuído, em específico o aprendizado federado, tem contribuído para revolucionar os serviços [Chamikara et al., 2021]. Liu *et al.* utilizam uma rede neural com unidades recorrentes de portas fechadas (*Gated Recurrent Unit – Neural Network* – GRU-NN) juntamente com o aprendizado federado (*Federated Learning* – FL) aplicados para solucionar o problema de predição de fluxo de tráfego [Liu et al., 2020].

Souza *et al.* propõem o *DFedForest*, um sistema de aprendizado federado para a criação distribuída e colaborativa de florestas aleatórias (RF) [Souza et al., 2020]. Os resultados mostram que o método aplicado teve uma acurácia maior em comparação aos modelos locais. O sistema aplica conceitos de aprendizado federado [McMahan et al., 2017], o que permite garantir a privacidade dos dados de treinamento, transmitindo apenas os modelos treinados. Outro ponto importante dessa abordagem é o compartilhamento dos modelos utilizando a tecnologia de cadeia de blocos (*blockchain*), o que garante a confiança mútua da rede e evita que participantes maliciosos prejudiquem o aprendizado do modelo. Contudo, o modelo introduz latência ao depender do compartilhamento de dados na cadeia de blocos.

O presente trabalho propõe o sistema de aprendizado de máquina distribuído com topologia par-a-par, utilizando um modelo compartilhado de Árvore de Decisão, em que as árvores compartilhadas compõem uma Floresta de Decisão Distribuída. O trabalho diferencia-se dos demais pois a agregação das Árvores de Decisão de cada nó participante ocorre sobre uma topologia de rede sobrecamada par-a-par. As árvores que compõem a Floresta de Decisão Distribuída garante a privacidade dos dados de treinamento do modelo ao mesmo tempo que fornecem um modelo global generalizado e com alta acurácia.

7. Conclusão

Este trabalho propôs um sistema de aprendizado de máquina distribuído baseado em comunicação par-a-par para a detecção de intrusão em redes. O sistema utiliza Árvores de Decisão como modelo de aprendizado de máquina e, quando agregadas pelo método *Bagging*, compõem uma Floresta de Decisão Distribuída. Os resultados da simulação do sistema mostram que o modelo de Floresta de Decisão Distribuída proposto atingiu uma mediana de acurácia de 79% nas mesmas condições utilizadas para a simulação do sistema de aprendizado federado baseado em topologia de servidor de parâmetros. A acurácia do modelo do sistema proposto ser inferior ao valor da acurácia do modelo do sistema com topologia de servidor de parâmetros pode ser explicada pois as Árvores de Decisão demandam uma maior quantidade de dados para atingirem desempenho ótimos em comparação às redes neurais, por outro lado necessitam de uma quantidade menor de processamento. Além disso, a topologia de comunicação par-a-par utilizada nesse sistema possui vantagens em comparação a topologia de servidor de parâmetros, pois mitiga os problemas de comunicação e ponto único de falha, já que os nós que compõem a rede se comunicam diretamente entre si e possuem as mesmas responsabilidades. Trabalhos futuros focam na realização de simulações com maior quantidade de dados e maior poder de processamento para a otimização do sistema proposto.

Agradecimentos

Este trabalho foi realizado com recursos do CNPq, FAPERJ, RNP, CAPES, CGI/FAPESP (2018/23062-5) e Prefeitura de Niterói/FEC/UFF (Edital PDPA 2020).

Referências

Alazab, M., RM, S. P., Parimala, M., Maddikunta, P. K. R., Gadekallu, T. R. e Pham, Q.-V. (2021). Federated learning for cybersecurity: Concepts, challenges, and future directions. *IEEE Transactions on Industrial Informatics*, 18(5):3501–3509.

Aragão, M. V. C., Mafra, S. B. e de Figueiredo, F. A. P. (2022). Análise de tráfego de rede com machine learning para identificação de ameaças a dispositivos IoT. Em *XL Simpósio Brasileiro de Telecomunicações e Processamento de Sinais (SBrT2022)*, Rio de Janeiro, RJ. SBrT.

Bellet, A., Guerraoui, R., Taziki, M. e Tommasi, M. (2018). Personalized and private peer-to-peer machine learning. Em Storkey, A. e Perez-Cruz, F., editors, *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research*, p. 473–481. PMLR.

Chamikara, M., Bertok, P., Khalil, I., Liu, D. e Camtepe, S. (2021). Privacy preserving distributed machine learning with federated learning. *Computer Communications*, 171:112–125.

Costa, L. H. M. K., de Amorim, M. D., Campista, M. E. M., Rubinstein, M. G., Florissi, P. e Duarte, O. C. M. B. (2012). Grandes massas de dados na nuvem: Desafios e técnicas para inovação. Em *Minicursos do SBRC 2012*, capítulo 1, p. 1–58. SBC, Porto Alegre.

Hard, A., Rao, K., Mathews, R., Ramaswamy, S., Beaufays, F., Augenstein, S., Eichner, H., Kiddon, C. e Ramage, D. (2018). Federated learning for mobile keyboard prediction. *arXiv preprint arXiv:1811.03604*.

Khan, M. A., Karim, M. R. e Kim, Y. (2019). A scalable and hybrid intrusion detection system based on the convolutional-lstm network. *Symmetry*, 11(4).

Lim, W. Y. B., Luong, N. C., Hoang, D. T., Jiao, Y., Liang, Y.-C., Yang, Q., Niyato, D. e Miao, C. (2020). Federated learning in mobile edge networks: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, 22(3):2031–2063.

Liu, C., Gu, Z. e Wang, J. (2021). A hybrid intrusion detection system based on scalable k-means+ random forest and deep learning. *IEEE Access*, 9:75729–75740.

Liu, H. e Lang, B. (2019). Machine learning and deep learning methods for intrusion detection systems: A survey. *Applied Sciences*, 9(20):4396.

Liu, Y., Yu, J. J. Q., Kang, J., Niyato, D. e Zhang, S. (2020). Privacy-preserving traffic flow prediction: A federated learning approach. *IEEE Internet of Things Journal*, 7(8):7751–7763.

Lopez, M. A., Silva, R. S., Alvarenga, I. D., Mattos, D. M. F. e Duarte, O. C. M. B. (2017). Coleta e caracterização de um conjunto de dados de tráfego real de redes de acesso em banda larga. Em *Anais do XXII Workshop de Gerência e Operação de Redes e Serviços*, Porto Alegre, RS, Brasil. SBC.

McMahan, B., Moore, E., Ramage, D., Hampson, S. e y Arcas, B. A. (2017). Communication-efficient learning of deep networks from decentralized data. Em *Artificial Intelligence and Statistics*, p. 1273–1282. PMLR.

Medeiros, D. S. V., Cunha Neto, H. N., Lopez, M. A., S. Magalhães, L. C., Fernandes, N. C., Vieira, A. B., Silva, E. F. e F. Mattos, D. M. (2020). A survey on data analysis

on large-scale wireless networks: online stream processing, trends, and challenges. *Journal of Internet Services and Applications*, 11(1).

Mothukuri, V., Parizi, R. M., Pouriyeh, S., Huang, Y., Dehghantanha, A. e Srivastava, G. (2021). A survey on security and privacy of federated learning. *Future Generation Computer Systems*, 115:619–640.

Neto, H. N. C., Dusparic, I., Mattos, D. M. F. e Fernande, N. C. (2022). FedSA: Accelerating intrusion detection in collaborative environments with federated simulated annealing. Em *2022 IEEE 8th International Conference on Network Softwarization (NetSoft)*, p. 420–428.

Neto, H. N. C., Mattos, D. M. F. e Fernandes, N. C. (2020). Privacidade do usuário em aprendizado colaborativo: Federated learning, da teoria à prática. Em *Minicursos do SBRC 2012*, capítulo 3, p. 101–155. Sociedade Brasileira de Computação, Porto Alegre, RS.

Sanz, I., Lopez, M. A., Rebello, G. A. e Duarte, O. C. (2018). Um sistema de detecção de ameaças distribuídas de rede baseado em aprendizagem por grafos. Em *Anais do XXXVI Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, p. 1187–1200, Porto Alegre, RS, Brasil. SBC.

Sapio, A., Canini, M., Ho, C.-Y., Nelson, J., Kalnis, P., Kim, C., Krishnamurthy, A., Moshref, M., Ports, D. R. K. e Richtárik, P. (2021). Scaling Distributed Machine Learning with In-Network Aggregation. Em *Proceedings of NSDI'21*.

Shi, S., Wang, Q. e Chu, X. (2018). Performance modeling and evaluation of distributed deep learning frameworks on GPUs. Em *2018 IEEE 16th Intl Conf on Dependable, Autonomic and Secure Computing, 16th Intl Conf on Pervasive Intelligence and Computing, 4th Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress(DASC/PiCom/DataCom/CyberSciTech)*, p. 949–957.

Souza, L., Rebello, G., Camilo, G., Guimarães, L. e Duarte, O. (2020). DFedForest: Floresta federada descentralizada. Em *Anais do XX Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais*, p. 355–368, Porto Alegre, Brasil. SBC.

Tang, F., Mao, B., Fadlullah, Z. M. e Kato, N. (2018). On a novel deep-learning-based intelligent partially overlapping channel assignment in SDN-IoT. *IEEE Communications Magazine*, 56(9):80–86.

Truong, N., Sun, K., Wang, S., Guitton, F. e Guo, Y. (2021). Privacy preservation in federated learning: An insightful survey from the GDPR perspective. *Computers & Security*, 110:102402.

Tuler De Oliveira, M., Reis, L. H. A., Verginadis, Y., Mattos, D. M. F. e Olabarriaga, S. D. (2022). Smartaccess: Attribute-based access control system for medical records based on smart contracts. *IEEE Access*, 10:117836–117854.