Abordagem baseada em Aprendizado Federado para a Detecção de Intrusão em Redes de Computadores

Alexsander Damaceno[1], **Maria do Rosário C. Ribeiro** [2,3], **Antonio Oliveira-Jr**[1,4] **e Renan R. de Oliveira**[1,5]

[1]Instituto de Informática – Universidade Federal de Goiás (UFG), GO, Brasil

[2]Faculdade de Informação e Comunicação (FIC) – Universidade Federal de Goiás (UFG), GO, Brasil

[3]Institute for Systems and Computer Engineering, Technology and Science (INESC-TEC), Porto, Portugal

[4]Fraunhofer Portugal AICOS, Porto, Portugal

[5]Instituto Federal de Goiás (IFG), GO, Brasil

alexdamaceno@discente.ufg.br, {rosarioribeiro,antoniojr}@ufg.br,

renan.rodrigues@ifg.edu.br

Abstract. *The rapid expansion of digital networks and the growing number of computer security incidents, the need for methods for preventing and detecting malicious activities becomes evident. Traditional approaches to network intrusion detection often face limitations in scalability, privacy, and adaptability. This article explores Federated Learning (FL) as a solution to address these challenges. By decentralizing the training process and preserving data privacy at the source, FL empowers network administrators to collaboratively build robust anomaly detection models without sharing sensitive information.*

Resumo. *A rápida expansão das redes digitais e o crescente número de incidentes de segurança computacional torna-se evidente a necessidade de métodos para a prevenção e detecção de atividades maliciosas. As abordagens tradicionais para detecção de intrusão em redes frequentemente enfrentam limitações em escalabilidade, privacidade e adaptabilidade. Este artigo explora o Aprendizado Federado (FL) como uma solução para lidar com esses desafios. Ao descentralizar o processo de treinamento e preservar a privacidade dos dados na fonte, o FL capacita os administradores de rede a construir colaborativamente modelos robustos de detecção de anomalias sem o compartilhamento de informações sensíveis.*

1. Introdução

O surgimento da era digital trouxe uma transformação profunda no cenário das redes de comunicação, trazendo consigo uma série de vantagens, mas também apresentando desafios substanciais em termos de segurança computacional. Com a expansão exponencial das redes, a detecção eficaz de anomalias tornou-se uma prioridade crucial para proteger ativos e informações críticas. Um Sistema de Detecção de Intrusões (*Intrusion Detection*

System - IDS) envolve a ação de monitorar os eventos que ocorrem em um sistema de computador ou em uma rede e examiná-los em busca de indícios de invasão.

Uma estratégias frequentemente adotadas na construção de um IDS é o treinamento de modelos de Aprendizado de Máquina (*Machine Learning* - ML) para serem utilizado na previsão de fluxos de dados que representam uma anomalia ou um tráfego normal do usuário. Neste caso, para treinar um modelo de ML para identificar e prevenir ataques em redes e sistemas de computadores, um IDS precisa operar realizando a coleta, o armazenamento e a análise de uma ampla gama de informações dos usuários [Niksefat et al. 2017].

Quando se trata do treinamento de modelos de ML, uma suposição é que estes modelos devem ser treinados centralmente na nuvem, usando dados de treinamento de vários dispositivos [Beutel et al. 2020]. No entanto, os dados refererentes ao tráfego da rede podem abranger informações sensíveis que podem representar um risco para a privacidade e a segurança das informações, como por exemplo, os endereços IPs, registros de atividade de rede, logs de eventos e pacotes de dados.

Dessa forma, as abordagens tradicionais de IDS enfrentam desafios de escalabilidade, privacidade e adaptação em tempo real. Neste contexto, o Aprendizado Federado (*Federated Learning* - FL) [McMahan et al. 2023] foi introduzido como uma abordagem de ML descentralizada que permite o treinamento colaborativo um modelo compartilhado, mantendo os dados sensíveis nos dispositivos. Nesta abordagem, somente os parâmetros dos modelos treinados localmente são compartilhados com o servidor agregador.

O FL é uma técnica de aprendizado de máquina descentralizada que permite a colaboração de várias partes interessadas, como dispositivos pessoais, organizações e servidores, com o objetivo de construir modelos de aprendizado de máquina sem o compartilhar dos dados brutos. Em vez disso, os modelos são treinados localmente nos dispositivos com seus próprios dados. Em seguida, apenas as informações agregadas e modelos atualizados são compartilhados entre as partes, protegendo a privacidade e a confidencialidade dos dados subjacentes [Veiga et al. 2023].

De acordo com [Yang et al. 2022], o FL apresenta algumas vantagens em redes comunicações, pois a trasmissão dos parâmetros do modelo de ML em vez dos dados de treinamento pode economizar energia, recursos de rede e latência da comunicação. Além do mais, o FL contribui com a preservação da privacidade dos dados, pois os dados de treinamento permanecem nos dispositivos.

Este trabalho aborda o uso do FL na detecção de intrusão em redes de computadores, apresentando uma visão de seus princípios fundamentais e aplicações práticas. Como principais contribuicões do artigo destaca-se a discussão das vantagens do FL em termos de acurácia, privacidade de dados e adaptação em tempo real, bem como, a apresentação de um estudo de caso que ilustram sua eficácia na detecção proativa de ameaças em ambientes de redes dinâmicas.

Para além desta seção introdutória, o restante deste artigo está organizado em seções, conforme descrito a seguir. A Seção 2 apresenta os trabalhos relacionados. A Seção 3 aborda os conceitos do FL. A Seção 4 descreve os elementos que compõem o cenário experimental. A Seção 5 discute os resultados da implementação de um sistema de detecção de intrusão em redes de computadores utilizando o FL e o FedAvg como

estratégia de agregação. Por fim, a Seção 6 apresenta as considerações finais.

2. Trabalhos Relacionados

Com o crescente número de incidentes de segurança computacional ao passar dos anos, métodos de prevenção e detecção de atividades maliciosas se fazem necessários. Em [dos Santos et al. 2018], os autores propõe a criação de um ambiente para a simulação de ataques computacionais, permitindo assim que os eventos de redes gerados por tais ataques possam ser utilizados para a avaliação de métodos de detecção de intrusão.

A utilização de FL também vem sendo adotado para a construção de sistemas de detecção de intrusão. Em [de Oliveira et al. 2023], os autores propõe um sistema IDS que usa a inteligência artificial federada e técnicas de privacidade diferencial, combinadas com comunicações assıncronas entre entidades do sistema, visando escalabilidade e confidencialidade dos dados. Os resultados mostraram que o modelo de detecção confidencial apresentam métricas satisfatorias de desempenho e, no caso de um ataque, é possível prever e determinar satisfatoriamente a sua natureza.

No trabalho de [Chen et al. 2020], os autores apresentam um modelo IDS utilizando FL aplicado em redes sem fio na borda. O sistema incorpora um conceito de mecanismo de atenção para avaliar a relevância dos parâmetros do modelo carregado. Em [Khan et al. 2021], os autores exploram a aplicação FL no contexto das redes de IoT com o objetivo de aprimorar as capacidades de detecção de intrusões, ao mesmo tempo que priorizam a preservação da privacidade dos dados.

A avaliacão desses trabalhos permitiu compreender os desafios para a implementação de métodos de detecção de intrusão. Por esta razão, este trabalho explora a implementação de um sistema IDS utizando FL em um ambiente baseado em contêineres Docker.

3. Aprendizado Federado

O FL é uma configuração de ML que permite que vários clientes (por exemplo, dispositivos móveis) treinem colaborativamente um modelo sem o compartilhamento dos dados locais com a coordenação de um servidor central [Amannejad 2020]. Existem duas entidades principais no processo FL: os proprietários dos dados, ou seja, os clientes, e o proprietário do modelo global, ou seja, o servidor.

Para uma conceituação do ambiente de FL, considere $S = \{k_1, k_2, ..., k_n\}$ como o conjunto de $\mathcal{N}$ clientes conectados um servidor antes do início de uma rodada de treinamento de FL. Cada cliente possui um conjunto de dados $\mathcal{P}_k$ com $n_k = |\mathcal{P}_k|$ amostras armazenadas em seus respectivos dispositivos locais. Nas abordagens clássicas de ML, todos os clientes enviam seus dados para um servidor e o servidor treina um modelo convencional, usando todos os dados $\mathcal{P} = \cup_{k=1}^{n} \mathcal{P}_k$. No entanto, isso não é possível para todos os cenários devido a questões de privacidade e limitações de largura de banda da rede.

No FL, os clientes treinam um modelo local usando seus próprios dados $\mathcal{P}_k$ e não enviam seus dados brutos para o servidor agregador. Neste caso, o que é enviado são apenas os parâmetros de cada modelo w treinado localmente. Os modelos locais recebidos pelo servidor são agregados para criar um modelo global w_{t+1} que é transmitido

aos clientes para uma nova rodada de treinamento. Este processo continua por t rodadas de comunicação [Li et al. 2021].

Em [Neto et al. 2020] é ilustrado o processo de treinamento que é adotado no FL. Conforme apresentado na Figura 1, no primeiro passo, o servidor envia o modelo inicial ao participantes do treinamento federado. Posteriormente, cada participante selecionado atualiza o modelo com seus dados locais e, por fim, o servidor agrega os parâmetros recebidos.

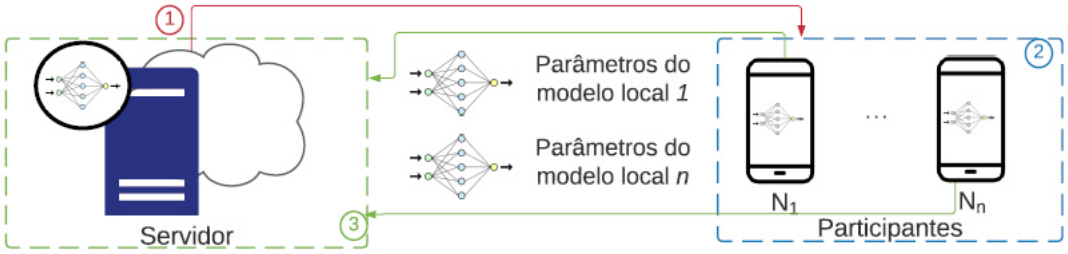

Figura 1. Arquitetura do FL [Neto et al. 2020].

O *Federated Averaging* (FedAvg) [McMahan et al. 2023] foi o primeiro algoritmo proposto para a agregação de modelos de FL. A ideia principal do FedAvg é realizar a agregação de um modelo global ao longo de várias rodadas de comunicação com base nos parâmetros dos modelos treinados localmente em cada dispositivo. No FedAvg, a agregação trata-se da média ponderada dos parâmetros dos clientes, cujos pesos são definidos com base na quantidade de dados utilizada no treinamento local.

4. Configuração dos Experimentos

Esta seção apresenta a descrição do conjunto de dados para a simulação de uma aplicação de IDS, o modelo de ML utilizado nos experimentos, bem como, o ambiente de simulação utilizado para a avaliação da convergência de tarefas de IDS em um ambiente federado.

4.1. Conjunto de Dados

Este trabalho utilizou um conjunto de dados denominado KDD-99 [KDD-99 1999] amplamente utilizado em pesquisas na área de IDS, disponibilizado publicamente pelo laboratório Lincoln do MIT. Este conjunto de dados foi escolhido por ser largamente utilizado em outros trabalhos da literatura para avaliar o desempenho de aplicações de IDS.

O KDD-99 foi gerado à partir do programa de Avaliação de Detecção de Intrusões da DARPA (*Defense Advanced Research Projects Agency*). A principal tarefa associada a este conjunto de dados é a detecção de intrusões. Ele é usado para treinar e avaliar modelos de aprendizado de máquina para detectar se o tráfego de rede representa comportamento normal ou vários tipos de ataques.

O conjunto de dados é conhecido por ser desbalanceado, com um grande número de conexões normais e um número relativamente pequeno de instâncias representando tipos específicos de ataques. Lidar com o desbalanceamento de classes é um desafio comum ao usar este conjunto de dados para aprendizado de máquina.

Os registros do conjunto de dados representam conexões, contendo 41 propriedades e aproximadamente 4,9 milhões de registros de conexão, dentre as quais tem-se: duração, protocolo, aplicação, flags, entre outros. Para cada conexão é atribuída uma etiqueta que indica sua classificação como "normal" ou o respectivo nome do ataque. Os ataques simulados possuem vários tipos que podem ser classificados como [dos Santos et al. 2018]:

- *Denial of Service* (DoS): Consiste na realização de inúmeras requisições a um recurso, de tal modo que o mesmo não é capaz de responder às requisições legítimas por estar totalmente ocupado.
- *User to Root* (U2R): Consiste na tentativa de obter acesso ao sistema como um usuário comum e depois escalar a hierarquia até chegar o nível de super-usuário.
- *Remote to Local* (R2L): Consite na capacidade de enviar pacotes para uma máquina que pode revelar vulnerabilidades que permitam ao atacante a exploração de vulnerabilidades locais para obter controle total da vítima.
- *Probing Attack* (Probe): Consiste na coleta de informações sobre o alvo, a fim de identificar possíveis vulnerabilidades.

Neste trabalho, a etapa de seleção de atributos foi baseada nos resultados do trabalho de [Kayacik et al. 2005] que utilizou uma abordagem baseada no ganho de informação para eliminar características irrelevantes para o processo de classificação. Após esta etapa, foram considerados apenas 27 propriedades para o treinamento dos modelos de ML.

4.2. Modelo de ML

O modelo de ML deste trabalho é apresentado na Figura 2. Este modelo é caracterizado por uma arquitetura de rede neural do tipo *Multi-Layer Perceptron* (MLP) com apenas uma camada oculta com 16 neurônios. Além do mais, é utilizado o ADAM (*Adaptive Moment Estimation*) como método de otimização do modelo, o *Sparse Categorical Crossentropy* como função de perda e um tamanho variado do *batch size*.

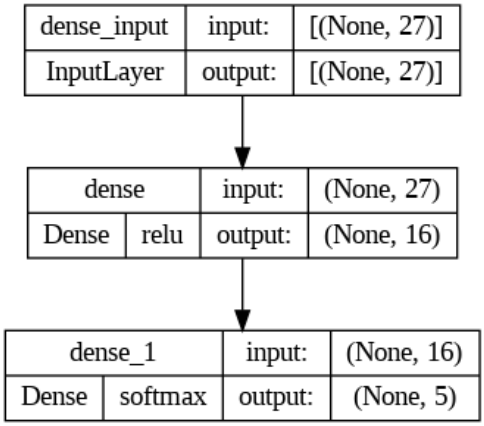

Figura 2. Arquitetura da Rede Neural

No contexto de FL, normalmente os dispositivos possuem recursos de rede, computação e armazenamento limitados, e portanto, o tamanho do modelo pode impactar no desempenho da tarefa. Neste trabalho, a arquitetura do MLP possui 533 parâmetros e um tamanho de 2.08 KB.

4.3. Ambiente de Simulação

A Figura 3 apresenta o ambiente de simulação deste trabalho. A orquestração do processo de FL é implementada utilizando o *framework* Flower [Beutel et al. 2020] em um ambiente baseado em contêineres Docker.

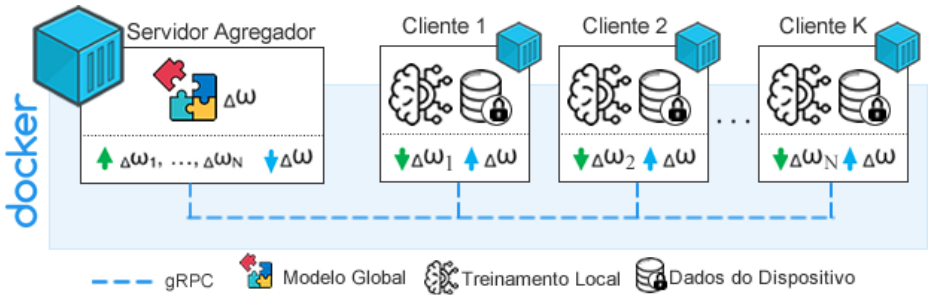

Figura 3. Ambiente de simulação de FL.

O Flower oferece uma implementação estável dos principais componentes de um sistema de FL, independente de linguagem de programação e de estrutura de ML. A utilização de contêineres permite o isolamento do servidor agregador e dos clientes participantes do processo de FL. Além do mais, é possível definir clientes heterogêneos personalizando limitações de computação e memória de cada contêiner, bem como, restringindo a largura de banda de rede utilizando ferramentas de controle de tráfego.

5. Discussão dos Resultados

Esta seção apresenta a discussão dos resultados da implementação de um sistema de detecção de instrusão em redes de computadores utilizando o FL e o FedAvg como estratégia de agregação. Inicialmente, na Figura 4 apresenta-se a evolução da acurácia de um modelo centralizado como forma de estabelecer um ponto de referência para avaliar qualitativamente a convergência de tarefas de FL para a detecção de intrusões. O modelo centralizado foi treinado por 100 épocas locais, utilizando 75% das amostras do conjunto de dados KDD-99 para o treinamento do modelo e 25% das amostras para o teste, atingindo uma acurácia no valor de 99.16%.

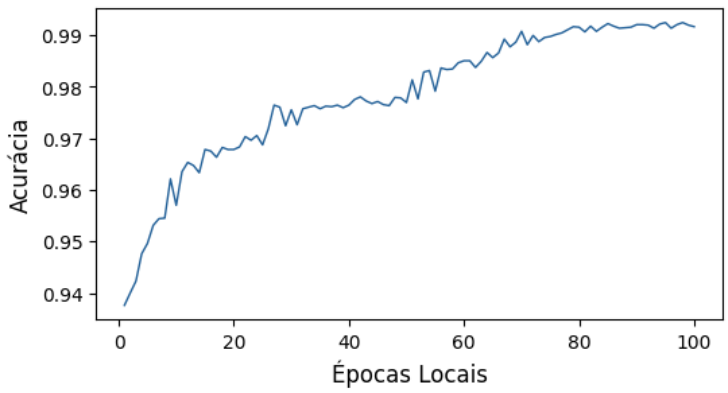

Figura 4. Treinamento do modelo centralizado.

Em seguida, foram realizados experimentos utilizando o paradigma FL variando três principais parâmetros: o número N de clientes participantes de cada rodada de treinamento, o tamanho B do batch utilizado durante o treinamento e o número E de épocas de treinamento local. Dessa forma, propõe-se analisar como esses fatores impactam a acurácia do modelo global da estratégia federada de detecção de intrusões. A Tabela 1 apresenta os parâmetros para análise de impacto de desempenho avaliados para as tarefas de FL para a detecção de intrusões.

Tabela 1. Parâmetros para análise de impacto do desempenho do FL.

Símbolo	Significado
N_t	Número total de clientes
N	Número de clientes participante de cada rodada de treinamento
B	Tamanho do batch de cada cliente
E	Número de épocas locais de treinamento

Para a geração dos conjuntos de dados locais dos clientes participantes do FL, as amostras do conjunto de dados KDD-99 foram embaralhadas e divididas em 20 conjuntos de dados com a mesma quantidade de exemplos, reservando 75% dos dados para o treinamento e 25% dos dados para o teste.

A Figura 5 apresenta a evolução da acurácia em cada rodada de comunicação do FL com N_t=20, E=1, B=16 e variações do número N de clientes participantes em cada rodada de treinamento. Após 100 rodadas de comunicação, todas as configurações alcançaram uma aproximação da acurácias do desempenho do modelo centralizado. Observa-se que, um pouco antes da rodada 40 o modelo global não atingiu uma estabilidade para o valor da acurácia. Neste caso, uma vez que o treinamento federado envolve a média ponderada dos modelos locais dos clientes, um grande número de modelos locais torna-se um desafio para o servidor agregador encontrar um ponto de convergência global que seja satisfatório para todos os clientes.

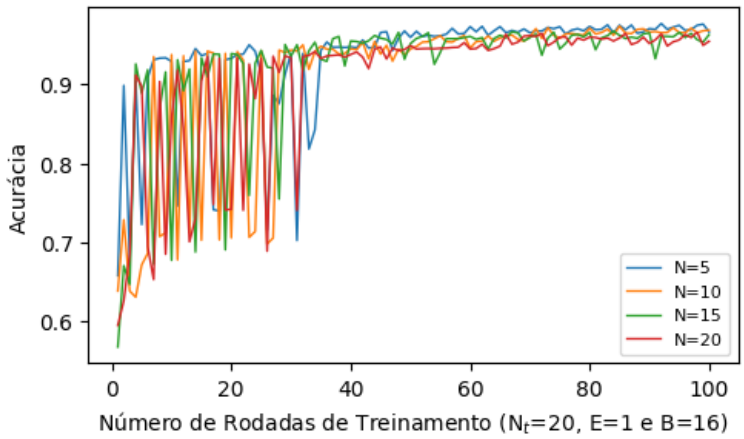

Figura 5. Acurácia do FL variando o número N de clientes de cada rodada.

Na Tabela 2 é possível observar que o modelo global de FL no cenário com variação do número de clientes obteve uma redução da acurácia de apenas -1.46%,

-1.79%, -1.42% e -2.54%. No entanto, é importante destacar que o objetivo do FL não é necessariamente replicar os resultados do treinamento centralizado. O FL deve ser avaliado em termos de sua capacidade para atingir uma solução de alta qualidade em cenários distribuídos, com o foco na preservação da privacidade e na eficiência de comunicação.

A Figura 6 apresenta a evolução da acurácia em cada rodada de comunicação do FL com N_t=20,N=5, B=16 e variações da quantidade E de épocas de treinamento local em cada rodada de treinamento. Na Tabela 2 pode-se observar que após a finalização das 100 rodadas de comunicação, a primeira configuração obteve uma acurácia de -0.81% com relação ao modelo centralizado. As demais configurações superaram a acurácia do modelo centralizado em $+0.17\%$, $+0.24\%$ e $+0.01\%$.

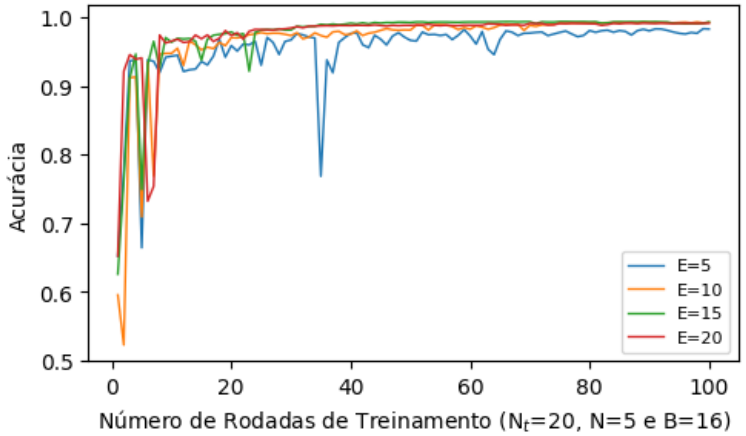

Figura 6. Acurácia do FL variando a quantidade E de épocas locais.

O resultado da variação do número de épocas locais supõe que o modelo centralizado pode atingir uma melhor acurácia desde que treinado por mais épocas. Além do mais, no contexto do FL onde os dados possuem a mesma distribuição, é possível afirmar que o treinamento tende a ocorrer por mais épocas quando comparado com o treinamento centralizado. Por exemplo, enquanto o modelo centralizado foi treinado por apenas 100 épocas, cada modelo local de FL chegou a ser treinado por até 5, 10, 15 e 20 épocas locais para cada uma das 100 rodadas de comunicação do treinamento distribuído.

A Figura 7 apresenta a evolução da acurácia em cada rodada de comunicação do FL com N_t=20,N=5, E=20 e variações do tamanho B do *batch* dos clientes. Na Tabela 2 pode-se observar que todas as configurações alcançaram uma aproximaram da acurácias do desempenho do modelo centralizado, obtendo uma redução de apenas -0.33%, -0.14%, -1.01% e -1.14%.

O resultado da variação do tamanho B do *batch* dos clientes indica um tendência da piora da acurácia à medida que B aumenta. O impacto do tamanho do lote no FL pode ser atribuído a várias razões, ainda mais quando um cliente tem poucos dados. Neste caso, um tamanho de lote muito grande pode resultar em uma amostra não que não refletem adequadamente a distribuição dos dados reais de todos os clientes.

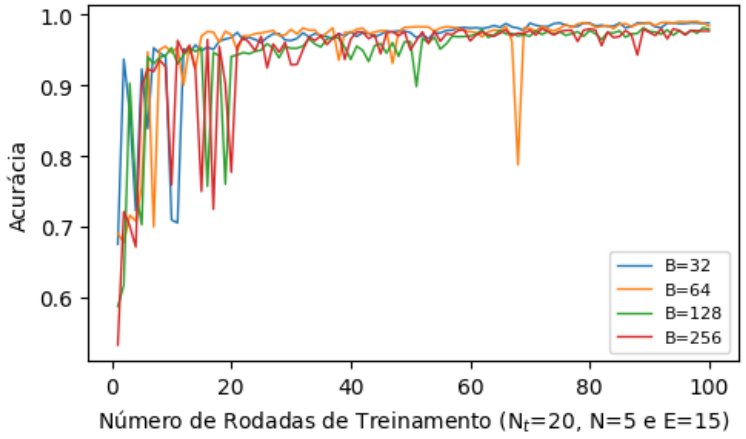

Figura 7. Acurácia do FL variando o tamanho B do *batch* dos clientes

Tabela 2. Avaliação do modelo global de FL, onde Acurácia$_D$ é a acurácia do modelo global e DIF$_{DC}$ é a diferença da Acurácia$_D$ com relação ao modelo centralizado.

N_t	N	E	B	**Acurácia$_D$ (%)**	**DIF$_{DC}$**
20	5	5	16	98.35	−0.81
		10		99.33	+0.17
		15		99.40	+0.24
		20		99.17	+0.01
20	5	1	16	97.70	−1.46
	10			97.37	−1.79
	15			96.74	−2.42
	20			96.62	−2.54
20	5	15	32	98.83	−0.33
			64	99.02	−0.14
			128	98.15	−1.01
			256	98.02	−1.14

6. Considerações Finais

O presente trabalho explorou o potencial do FL para implementação de um IDS utilizando o conjuto de dados KDD-99 e o *framework* Flower para a orquestração do processo de FL em um ambiente baseado em contêineres Docker. Os experimentos e análises revelaram como o FL pode ser usado em cenários de segurança da informação, bem como sobre a influência de fatores-chave, como o número de clientes, o tamanho do *batch* e o número de épocas locais de treinamento. Dessa forma, este estudo demonstrou o potencial do FL e aprendizado federado para tarefas de detecção de intrusões em redes de computadores distribuídas e que pode ser obter um bom resultado com esse técnica. Esta pesquisa contribui para a compreensão e aplicação do FL em ambientes distribuídos.

Referências

Amannejad, Y. (2020). Building and Evaluating Federated Models for Edge Computing. In *2020 16th International Conference on Network and Service Management (CNSM)*,

pages 1–5.

Beutel, D. J., Topal, T., Mathur, A., Qiu, X., Parcollet, T., and Lane, N. D. (2020). Flower: A Friendly Federated Learning Research Framework. *CoRR*, abs/2007.14390.

Chen, Z., Lv, N., Liu, P., Fang, Y., Chen, K., and Pan, W. (2020). Intrusion Detection for Wireless Edge Networks based on Federated Learning. *IEEE Access*, 8:217463–217472.

de Oliveira, J. A., Meneguette, R. I., Gonçalves, V. P., de Sousa Jr, R. T., Guidoni, D. L., Oliveira, J. C., and Rocha Filho, G. P. (2023). F-NIDS–Sistema de detecçao de Intrusao descentralizado com base em Aprendizado Federado. In *Anais do XLI Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, pages 29–42. SBC.

dos Santos, L. L. S., Machado, R. B., and dos Santos Vieira, G. (2018). Proposta de Ambiente Experimental para Avaliação de Sistemas de Detecção de Intrusão. *BTSYM 2018 Proceedings*.

Kayacik, H., Zincir-Heywood, A.N., and Heywood, M. (2005). Selecting Features for Intrusion Detection: A Feature Relevance Analysison KDD 99. *In Proceedings of the Third Annual Conference on Privacy, Security and Trust*.

KDD-99 (1999). Kdd Cup 1999 Data. disponível em: `https://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html`.

Khan, L. U., Saad, W., Han, Z., Hossain, E., and Hong, C. S. (2021). Federated Learning for Internet of Things: Recent Advances, Taxonomy, and Open Challenges. *IEEE Communications Surveys & Tutorials*, 23(3):1759–1799.

Li, Q., Wen, Z., Wu, Z., Hu, S., Wang, N., Li, Y., Liu, X., and He, B. (2021). A Survey on Federated Learning Systems: Vision, Hype and Reality for Data Privacy and Protection. *IEEE Transactions on Knowledge and Data Engineering*, PP:1–1.

McMahan, H. B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. (2023). Communication-Efficient Learning of Deep Networks from Decentralized Data. *arXiv*.

Neto, H. N. C., Mattos, D. M. F., and Fernandes, N. C. (2020). Privacidade do Usuário em Aprendizado Colaborativo: Federated Learning, da Teoria à Prática. In *Livro de Minicursos do SBSEG 2020. Sociedade Brasileira de Computação.* SBC.

Niksefat, S., Kaghazgaran, P., and Sadeghiyan, B. (2017). Privacy Issues in Intrusion Detection Systems: A Taxonomy, Survey and Future Directions. *Computer Science Review*, 25:69–78.

Veiga, R., Both, C. B., Medeiros, I., Rosário, D., and Cerqueira, E. (2023). A Federated Learning Approach for Authentication and User Identification based on Behavioral Biometrics. In *Anais do XLI Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, pages 15–28. SBC.

Yang, Z., Chen, M., Wong, K.-K., Poor, H. V., and Cui, S. (2022). Federated Learning for 6G: Applications, Challenges, and Opportunities. *Engineering*, 8:33–41.