FedPEAC-Net: Seleção de Atributos e Modelo Neural Híbrido Baseados em Autocodificadores para Sistemas de Detecção de Intrusão Federados*

Ricardo A. Lundgren[1], **Tadeu N. Ferreira**[2] **e Diogo M. F. Mattos**[1]

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

{ricardolundgren, tadeu_ferreira, diogo_mattos}@id.uff.br

Abstract. *The accelerated growth of the Internet and smart devices has increased the volume of data generated at the network edge, also raising the incidence of attacks targeting the exploitation of sensitive information. Thus, the relevance and necessity of distributed, efficient Intrusion Detection Systems (IDS) that preserve user privacy become evident. This paper proposes FedPEAC-Net, an IDS based on federated learning that combines an autoencoder and a binary classifier trained in parallel, integrated with a feature selection technique guided by reconstruction error. The adopted methodology involves experimentation with the CICIDS2017 dataset in a federated scenario with 10 clients. The results show that FedPEAC-Net achieves an F1-score of 0.95 before and 0.93 after feature selection, showing only a 2.3% drop, while ResNet and LeNet-5 models showed reductions of 15.6% and 15%, respectively. These results confirm the effectiveness of the proposed approach in distributed environments, even with resource-constrained devices.*

Resumo. *O crescimento acelerado da Internet e de dispositivos inteligentes ampliou o volume de dados gerados na borda da rede, elevando também a incidência de ataques voltados à exploração de informações sensíveis. Assim, tornam-se evidentes a relevância e a necessidade de sistemas de detecção de intrusão (Intrusion Detection Systems - IDS) distribuídos, eficientes e que preservem a privacidade dos usuários. Este artigo propõe o FedPEAC-Net, um IDS baseado em aprendizado federado que combina um autocodificador e um classificador binário treinados em paralelo, integrados a uma técnica de seleção de atributos guiada pelo erro de reconstrução. A metodologia adotada envolve experimentação com o conjunto de dados CICIDS2017 em um cenário federado com 10 clientes. Os resultados demonstram que o FedPEAC-Net alcança F1-score de 0,95 antes e 0,93 após a seleção de atributos, mostrando redução de apenas 2,3%, enquanto modelos ResNet e LeNet-5 apresentaram quedas de 15,6% e 15%. Esses resultados confirmam a eficácia da proposta para ambientes distribuídos, mesmo com dispositivos com recursos limitados.*

1. Introdução

O crescimento exponencial da Internet e a ampla adoção de dispositivos inteligentes e móveis, como *smartphones*, veículos conectados e dispositivos da Internet das

*Este trabalho foi realizado com recursos do CNPq, CAPES, FAPERJ, RNP, Prefeitura de Niterói/FEC/UFF (Edital PDPA 2020) e INCT ICONIOT.

Coisas (*Internet of things* - IoT), impulsionam o aumento do volume de dados gerados e processados na borda da rede. Nesse novo cenário, informações sensíveis dos usuários ficam armazenadas localmente, tornando esses dispositivos alvos atrativos para ciberataques. Segundo relatório da Kaspersky, na América Latina, houve um aumento de 70% no número de tentativas de ataques contra plataformas móveis entre os biênios de 2022-2023 e 2023-2024, destacando-se Brasil, México e Equador como os países mais afetados[1]. A crescente incidência desses ataques compromete a segurança de dados, causa prejuízos financeiros e afeta a operação de serviços essenciais, como saúde e governo digital. Como resposta, estratégias automatizadas de detecção de intrusões passaram a desempenhar papel central na defesa da cibersegurança, sobretudo quando integradas a técnicas de aprendizado de máquina [Bout et al., 2022, Abdulganiyu et al., 2023].

Dentre abordagens promissoras para o desenvolvimento de Sistemas de Detecção de Intrusão (*Intrusion Detection Systems* – IDS), destaca-se o uso de aprendizado federado, que viabiliza o treinamento colaborativo de modelos de aprendizado profundo em dispositivos locais, preservando a privacidade dos dados. Essa técnica, entretanto, enfrenta desafios como a heterogeneidade computacional entre os clientes participantes e a elevada dimensionalidade dos dados, que pode comprometer a eficiência e escalabilidade do modelo global [Cunha Neto et al., 2020, Neto et al., 2023]. Assim, a redução da dimensionalidade surge como uma etapa essencial, especialmente em ambientes distribuídos, visando diminuir a sobrecarga de processamento sem perda de desempenho na detecção de ataques [Andreoni Lopez et al., 2019, Barbosa et al., 2024].

Este artigo propõe o FedPEAC-Net, um modelo de detecção de intrusão baseado em aprendizado federado que combina uma rede neural híbrida composta por um autocodificador (*autoencoder*) e um classificador binário treinados paralelamente (*Federated Parallel Encoder-Autoencoder-Classifier Network* - FedPEAC-Net), e uma técnica de seleção de atributos baseada na minimização do erro de reconstrução. O objetivo é reduzir a dimensionalidade dos dados de entrada, mitigando o impacto da heterogeneidade entre os dispositivos e mantendo a acurácia do modelo. A proposta foi avaliada com o conjunto CICIDS2017 [Sharafaldin et al., 2018], simulando um ambiente federado com 10 clientes. O treinamento e os testes foram conduzidos com dados reais, buscando validar a eficácia do FedPEAC-Net tanto em desempenho quanto em viabilidade prática.

Em comparação com trabalhos anteriores que aplicam autocodificadores combinados com classificadores tradicionais como máquinas de vetores de suporte (*Support Vector Machine* – SVM) [Kunang et al., 2018] ou utilizam métodos clássicos de seleção de atributos baseados em heurísticas [Nimbalkar e Kshirsagar, 2021], esta proposta apresenta duas inovações principais: (i) um modelo inteiramente neural com treinamento paralelo entre o codificador e o classificador, e (ii) uma técnica de seleção de atributos que considera diretamente o erro de reconstrução como critério. Os resultados experimentais mostram que o FedPEAC-Net alcança medida F1 de 0,95 antes e 0,93 após a seleção de atributos, mostrando uma redução de apenas 2,3%, enquanto modelos tradicionais, como ResNet e LeNet-5, apresentaram quedas de 15,6% e 15%. Os resultados então mostram que a proposta é robusta para ambientes com restrições de recursos, superando abordagens existentes em termos de eficiência e desempenho na detecção de intrusões.

[1]Disponível em `https://www.kaspersky.com.br/blog/panorama-ameacas-latam-2024/22888/`.

O restante do artigo está organizado da seguinte forma. A Seção 2 elenca os trabalhos relacionados. O problema relacionado à dimensionalidade em um cenário federado é analisado na Seção 3. O modelo híbrido e a seleção de atributos são propostos na Seção 4. A Seção 5 avalia a proposta e discute os resultados obtidos. A Seção 6 conclui o trabalho.

2. Trabalhos Relacionados

Trabalhos anteriores exploram estratégias para lidar com os desafios impostos pela alta dimensionalidade dos dados e pela heterogeneidade dos dispositivos em sistemas de detecção de intrusão, especialmente em ambientes distribuídos como redes IoT. As principais abordagens incluem o uso de técnicas de seleção de atributos, autocodificadores para extração de características e a adoção de aprendizado federado visando preservar a privacidade dos dados. Nimbalkar e Kshirsagar propõem uma técnica de seleção de atributos baseada em métricas de informação (*Information Gain* e *Gain Ratio*), selecionando os 50% atributos mais relevantes [Nimbalkar e Kshirsagar, 2021]. Essa abordagem, aplicada aos conjuntos de dados BoT-IoT e KDD Cup 1999, demonstrou ganhos em desempenho e redução do tempo de treinamento com o classificador JRip. O foco da proposta consiste a filtragem de atributos redundantes para permitir modelos mais leves e eficientes.

Hussain *et al.* propõem uma arquitetura derivada do ResNet para detecção de ataques de negação de serviço (*Denial of Service* - DoS) em redes IoT [Hussain et al., 2020]. Utilizando o conjunto BoT-IoT, a rede treinada obteve desempenho elevado, com acurácia de 99,99% para ataques DoS e 87% na classificação multiclasse. A abordagem destaca a robustez de arquiteturas profundas, embora sem considerar estratégias de redução de dimensionalidade e a generalização da proposta. Por sua vez, Cui *et al.* modificam a arquitetura LeNet-5, resultando no modelo LeNet-4, com menor complexidade e alta acurácia na detecção de anomalias [Cui et al., 2020]. O modelo incorpora uma etapa de seleção de atributos por eliminação recursiva, baseada em florestas aleatórias, e alcança acurácia superior a 98% na classificação binária usando o conjunto CICIDS2017.

Kunang *et al.* apresentam um modelo que integra um autocodificador com uma máquina de vetores de suporte (*Support Vector Machine* - SVM), executando ambas as etapas em paralelo para reduzir a dimensionalidade e classificar os dados de forma eficiente [Kunang et al., 2018]. A proposta atinge bons resultados utilizando funções de ativação ReLU e custo de entropia cruzada, destacando-se pela combinação de técnicas clássicas e redes neurais. Qin e Kondo investigam a aplicação do aprendizado federado em IDS, utilizando um algoritmo ganancioso para selecionar subconjuntos de atributos específicos para cada tipo de ataque [Qin e Kondo, 2021]. A proposta emprega a arquitetura ONLAD, baseada em autocodificador e OS-ELM, e demonstra que modelos especializados por tipo de ataque, aliados à seleção personalizada de atributos, resultam em maior acurácia no conjunto de dados NSL-KDD. Cunha Neto *et al.* propõem o método FedSBS (*Federated Score-Based Selection*), que combina aprendizado federado e seleção de participantes para sistemas de detecção de intrusão, enfrentando o desafio de dados não uniformes e presença de participantes maliciosos. O método utiliza um sistema de pontuação baseado em ganho de informação, seleção *epsilon-greedy* e agregação com momento global para melhorar a estabilidade do modelo. Experimentos mostraram que o FedSBS alcança desempenho superior às abordagens tradicionais, com precisão de 90% e F1-score de 80%, mesmo em situações adversas com até 60% de participantes maliciosos, destacando sua robustez em cenários críticos de segurança cibernética [Cunha Neto et al., 2024].

Embora os trabalhos anteriores demonstrem avanços na aplicação de aprendizado federado profundo e seleção de atributos em IDS, os trabalhos adotam abordagens sequenciais ou especializadas, dependentes de algoritmos externos ou modelos híbridos com classificadores tradicionais. Em contraste, este artigo propõe um modelo totalmente baseado em redes neurais, o FedPEAC-Net, que realiza o treinamento paralelo entre autocodificador e classificador binário, com seleção de atributos baseada no erro de reconstrução. A proposta permite um processo de extração de características e classificação coeso, eficiente e mais robusto à heterogeneidade de dispositivos, mantendo desempenho elevado.

3. Problema da Dimensionalidade em Sistemas de Detecção de Intrusão no Aprendizado Federado

O uso da técnica de aprendizado federado em sistemas de detecção de intrusão permite o treinamento de modelos de inteligência artificial sem a centralização dos dados, promovendo maior privacidade e segurança para os dispositivos participantes. A abordagem distribui os requisitos computacionais entre os clientes da rede, favorecendo a escalabilidade e a eficiência em ambientes heterogêneos. Entretanto, a aplicação do aprendizado federado em tarefas de classificação em IDS enfrenta limitações decorrentes da alta dimensionalidade dos dados. Em particular, o grande número de atributos necessários para descrever os diferentes cenários de tráfego de rede impõe desafios à capacidade dos modelos em generalizar e operar de forma eficiente em dispositivos com recursos limitados.

A presença de um número excessivo de atributos, ou características, pode causar sobreajuste (*overfitting*), dificultar a adaptação a novos dados e comprometer a capacidade de generalização dos modelos. A complexidade computacional, por sua vez, cresce com o aumento da dimensionalidade, elevando o custo de processamento sem, necessariamente, melhorar o desempenho do sistema [Meng et al., 2017, Barbosa et al., 2024]. As abordagens para redução de dimensionalidade são, em geral, divididas entre extração e seleção de atributos. A extração gera um novo conjunto de atributos combinando os existentes, enquanto a seleção identifica subconjuntos relevantes do conjunto original. Ambas as estratégias visam reduzir o número de atributos, preservando aqueles mais importantes para a tarefa de detecção [Meng et al., 2017, Barbosa et al., 2024].

A aplicação de seleção de atributos no contexto de aprendizado federado busca mitigar problemas como o alto custo computacional, baixa generalização e heterogeneidade de desempenho entre os clientes. Em redes com nós distribuídos, a diversidade de hardware entre os dispositivos leva a diferentes capacidades de processamento nos treinos locais [Cunha Neto et al., 2024]. Sistemas de Detecção de Intrusão baseados em redes neurais e com elevado número de atributos enfrentam maior complexidade computacional, o que pode inviabilizar sua execução em dispositivos restritos [Kasongo e Sun, 2020]. A integração da seleção de atributos com o aprendizado federado adiciona ainda desafios relacionados à heterogeneidade dos dados locais. Como os subconjuntos ideais de atributos podem variar entre os clientes, surgem dificuldades na etapa de agregação dos modelos locais em um modelo global coerente.

Diante desse cenário, torna-se evidente a necessidade de estratégias que aliem redução de dimensionalidade e aprendizado federado de forma eficiente e adaptável às limitações dos dispositivos participantes. A adoção de técnicas capazes de selecionar ou

extrair atributos relevantes, mantendo a coesão entre os modelos locais e o modelo global, é fundamental para garantir a viabilidade de IDS em ambientes federados e heterogêneos. Assim, o desenvolvimento de soluções que considerem simultaneamente a eficiência computacional, a preservação da privacidade e a robustez dos modelos representa uma direção promissora para o avanço da segurança em redes, principalmente redes móveis e IoT.

4. FedPEAC-Net e a seleção de atributos por minimização do erro de reconstrução

O FedPEAC-Net (*Federated Parallel Encoder-Autoencoder-Classifier Network*) é uma arquitetura híbrida aplicada a sistemas de detecção de intrusão em ambiente federado. Nesse contexto, múltiplos clientes mantêm os dados localmente, treinando modelos com um autocodificador (*autoencoder*) e um classificador binário em paralelo. Após cada rodada, os parâmetros locais são agregados por um servidor central para formar o modelo global. A estrutura visa: (i) reduzir as perdas na reconstrução dos dados de entrada e (ii) aumentar a eficiência na detecção de intrusões em tráfego de rede. A estrutura busca extrair representações compactas e informativas, mantendo a capacidade de classificação precisa, mesmo em cenários de alta dimensionalidade, comuns em sistemas de detecção de intrusão. A Figura 1 apresenta o diagrama de blocos da arquitetura proposta.

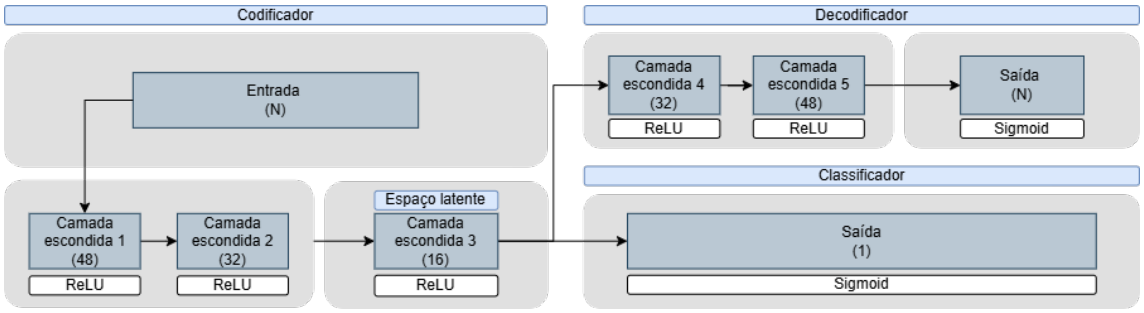


Figura 1. Arquitetura da rede neural híbrida FedPEAC-Net. A entrada com N atributos passa por três camadas densas no codificador, formando o espaço latente compartilhado. Esse espaço alimenta simultaneamente o decodificador, que reconstrói os dados originais, e o classificador binário, responsável por prever a presença de intrusões.

O autocodificador inicia com uma camada de entrada contendo N neurônios, sendo $N = 77$ para o caso de uso do conjunto de dados CICIDS2017, sem seleção de atributos e $N = 58$ com seleção de atributos. A menor camada do autocodificador, o espaço latente, representa as características mais relevantes dos dados, otimizando a reconstrução no decodificador, que realiza o processo inverso até retornar à dimensão original. O treinamento do autocodificador visa minimizar a perda de reconstrução com base na função de custo do Erro Quadrático Médio (MSE):

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2, \tag{1}$$

em que n representa o número de amostras, y_i é o valor real da i-ésima amostra e $\hat{y}_i$ é o valor reconstruído.

O classificador binário recebe como entrada a saída do gargalo (espaço latente) do autocodificador e é responsável por classificar a saída como intrusão ou não. O treinamento do classificador utiliza a função de custo de Entropia Cruzada Binária (*Binary Cross Entropy* - BCE), dada por:

$$BCE = -\frac{1}{n}\sum_{i=1}^{n}[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)], \tag{2}$$

em que y_i representa o rótulo real da i-ésima amostra e $\hat{y}_i$ a probabilidade prevista para a classe positiva.

A função de custo total utilizada para o treinamento conjunto da arquitetura é uma combinação ponderada das perdas do autocodificador e do classificador, expressa por:

$$Loss = \alpha \cdot MSE + \beta \cdot BCE, \tag{3}$$

com os coeficientes $\alpha = 0,5$ e $\beta = 1,0$, assegurando o balanceamento entre reconstrução e classificação durante o treinamento de forma a priorizar a classificação.

A arquitetura é treinada utilizando o otimizador Adam, escolhido por sua eficiência em lidar com dados desbalanceados e por sua rápida convergência em comparação ao Gradiente Descendente Estocástico (*Stochastic Gradient Descent* - SGD). Para a avaliação do desempenho do modelo, são utilizadas a métrica $F1$, por considerar o equilíbrio entre precisão e revocação, e a AUC-PR, área sob a curva de precisão e revocação, mais apropriada do que a AUC-ROC em cenários com classes desbalanceadas.

A seleção de atributos proposta tem como finalidade avaliar o impacto da escolha de características com menor erro de reconstrução no desempenho da arquitetura FedPEAC-Net, bem como em modelos comparativos. Para isso, utiliza-se um autocodificador simples, ilustrado na Figura 2, composto por uma camada de entrada e uma de saída com 77 neurônios em cada e uma camada oculta com 16 neurônios. A camada oculta utiliza a função de ativação ReLU, que introduz não linearidades ao modelo, enquanto a camada de saída adota ativação linear, favorecendo a reconstrução precisa dos dados. A função de custo utilizada é o Erro Quadrático Médio (*Mean Square Error* - MSE), buscando minimizar o erro entre os dados de entrada e saída. O otimizador Adam é mantido nesta configuração por sua robustez e adaptabilidade. Após o treinamento, os atributos são ranqueados com base em seus erros de reconstrução e os pertencentes ao percentil 75,

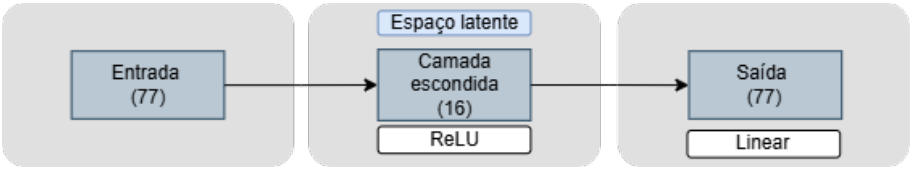


Figura 2. Arquitetura do autocodificador simples utilizado na seleção de atributos. A entrada possui 77 atributos, que são comprimidos em uma camada oculta de 16 neurônios com ativação ReLU. A camada de saída reconstrói os dados com 77 neurônios. O erro de reconstrução (MSE) é utilizado para ranquear os atributos com base em sua relevância.

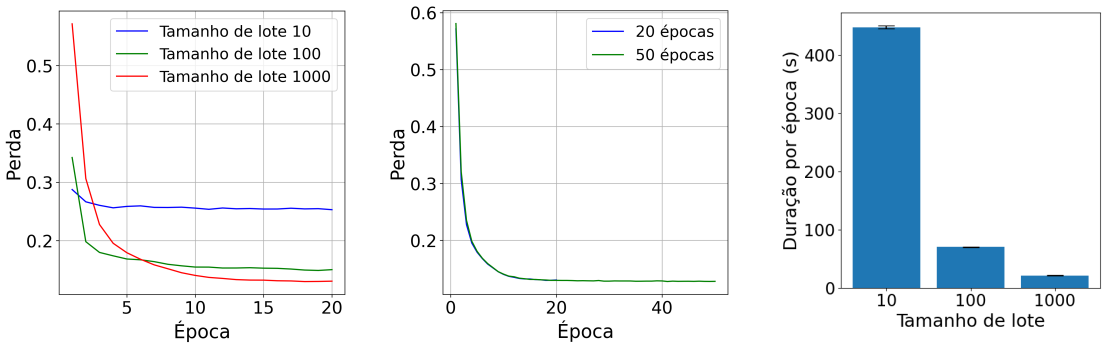
com os menores erros, são selecionados. Essa abordagem visa identificar os atributos com maior poder de reconstrução após a compressão, otimizando a representação dos dados para os modelos avaliados.

A integração dos componentes, autocodificador, classificador binário e mecanismo de seleção de atributos, constitui uma abordagem coesa, eficiente e adaptada a ambientes federados com dispositivos heterogêneos. O treinamento paralelo entre reconstrução e classificação favorece a aprendizagem de representações latentes robustas, enquanto a seleção de atributos orientada pelo erro de reconstrução contribui para a redução da dimensionalidade com mínima perda de informação. A seção de resultados apresenta uma avaliação quantitativa da proposta, considerando diferentes cenários e comparando o desempenho obtido frente a outras arquiteturas.

5. Avaliação e Resultados

O conjunto de dados utilizado para avaliar a técnica de seleção de atributos e o modelo híbrido foi o CICIDS2017 [Sharafaldin et al., 2018]. O conjunto simula tráfego de rede real e inclui tanto fluxos benignos quanto maliciosos, abrangendo ataques como *Brute Force FTP*, *Brute Force SSH*, *DoS*, *Heartbleed*, *Web Attack*, *Infiltration*, *Botnet* e *Distributed Denial of Service* (DDoS) [Silva et al., 2022]. O CICIDS2017 é amplamente adotado para treinamento e avaliação de IDS. Trata-se de um conjunto desbalanceado, no qual aproximadamente 20% dos fluxos são rotulados como ataques. Essa característica reflete um cenário realista, em que o tráfego malicioso representa uma minoria em relação aos dados benignos. A presença de 16 tipos distintos de ataques contribui para a generalização dos modelos de classificação. O conjunto conta com cerca de três milhões de registros e 79 atributos por entrada. O atributo categórico *Destination Port* foi removido do experimento devido à elevada cardinalidade, que tornaria os dados esparsos e dificultaria o treinamento eficiente dos modelos. A configuração experimental baseia-se em uma rede neural que recebe como entrada fluxos de rede benignos e maliciosos, com o objetivo de classificar intrusões enquanto reconstrói os dados por meio de um autocodificador. Os dados foram submetidos a pré-processamento, utilizando a normalização *MinMaxScaler*, da biblioteca `scikit-learn`, escalando os valores para o intervalo $[0, 1]$. Posteriormente, o conjunto foi dividido em dados de treino (70%) e teste (30%), permitindo a avaliação consistente do desempenho dos modelos.

Considerando a aplicação em aprendizado federado, os dados foram distribuídos de forma aleatória entre 10 clientes, sem sobreposição entre eles. A divisão foi feita de maneira não uniforme, com diferentes quantidades de dados atribuídas a cada cliente, simulando um cenário realista de heterogeneidade em ambientes federados. A arquitetura adotada inclui um servidor central e 10 clientes. O servidor orquestra o treinamento federado, agregando os pesos dos modelos locais e atualizando o modelo global com o uso da estratégia de Média Federada (*Federated Averaging*). Os clientes, por sua vez, treinam seus modelos com os dados locais e compartilham os parâmetros atualizados ao final de cada rodada. A comunicação e coordenação entre servidor e clientes são viabilizadas pelo arcabouço Flower [Wang et al., 2022], que permite a simulação de múltiplos clientes em uma única máquina, além de oferecer suporte agnóstico a diferentes bibliotecas de aprendizado de máquina. O servidor é responsável por configurar a porta de conexão, definir o número de rodadas de treinamento, controlar o fluxo de execução e especificar a estratégia de agregação. Cada cliente conecta-se ao servidor, define hiperparâmetros



(a) Curvas de perda com 20 épocas.

(b) Curvas de perda com 50 épocas.

(c) Tempo médio de treino.

Figura 3. Avaliação empírica dos parâmetros de treinamento para o autocodificador simples utilizado na seleção de atributos.

locais como número de épocas e tamanho de lote, realiza o treinamento com os dados locais e envia os pesos ao servidor ao final de cada rodada. Os clientes também realizam predições e avaliações locais com seus respectivos modelos.

Os experimentos foram realizados em ambiente Ubuntu 24.04 com Python (versão 3.9.10) que incluiu as bibliotecas: `pandas` (2.2.2), `numpy` (1.23.5), `scikit-learn` (1.5.1), `keras` (2.15.0), `tensorflow` (2.15.0) e `flwr` (1.10.0), em um computador com CPU i7-13700KF e 128 GB RAM. Os experimentos avaliam a seleção de atributos proposta e o impacto da seleção nos modelos ResNet, LeNet-5 e FedPEAC-Net. Para garantir validade estatística, cada experimento foi repetido 10 vezes. Os resultados apresentados são médias e a distribuição *t-student* foi empregada para o cálculo dos intervalos de confiança de 95%.

Para definir os melhores parâmetros de treinamento para a etapa de seleção de atributos, foi realizada uma busca empírica por meio da variação do tamanho de lote e, posteriormente, do número de épocas. Inicialmente, adotou-se o tamanho de lote igual a 10, considerado pequeno [Masters e Luschi, 2018], com o objetivo de observar a ordem de grandeza que proporciona uma curva de perda satisfatória. A escolha dos valores foi inspirada em uma estratégia semelhante à busca binária, visando encontrar aproximações eficientes para o parâmetro ideal. Inicialmente, 20 épocas foram utilizadas como ponto de partida para avaliar a ordem de grandeza necessária ao treinamento do autocodificador simples. No entanto, os experimentos demonstraram que esse valor não era suficiente para garantir a convergência da função de custo, conforme ilustrado na Figura 3(b). Após a definição do melhor tamanho de lote, a variação no número de épocas teve como objetivo refinar esse parâmetro, buscando garantir o aprendizado adequado do modelo com o menor custo computacional possível. Os testes realizados consideraram quatro diferentes cenários, nos quais foram combinadas variações de tamanho de lote e número de épocas, visando identificar a configuração mais eficiente para o treinamento do autocodificador simples. A análise levou em conta tanto a convergência da função de custo quanto o tempo de execução por época.

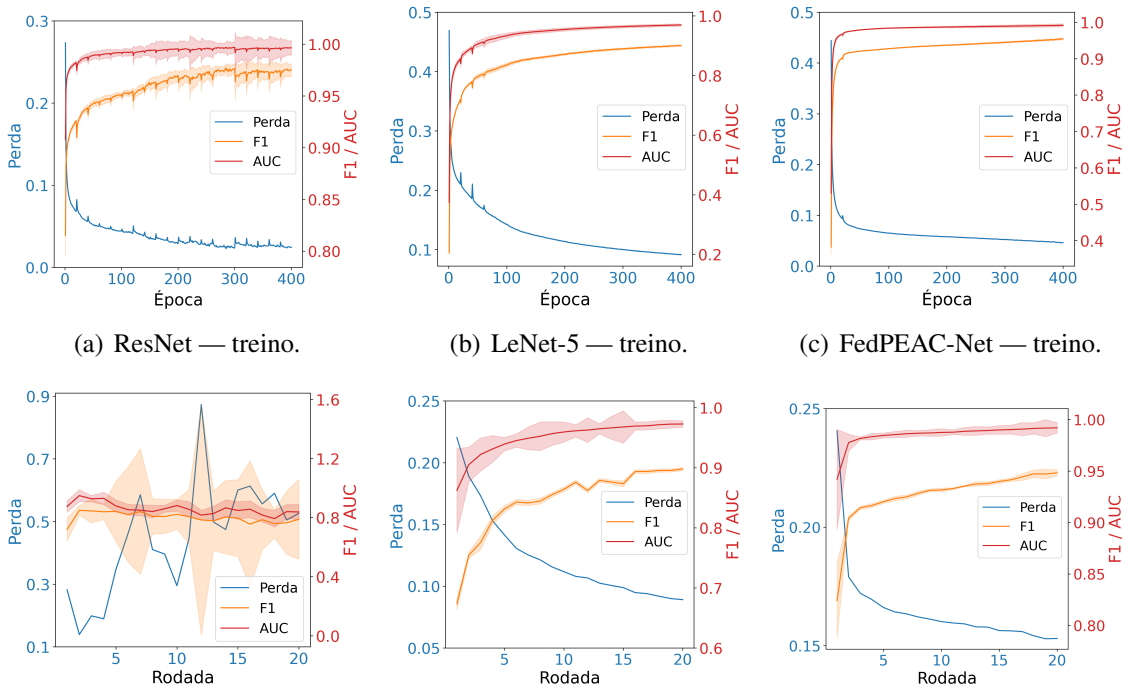
A métrica de avaliação adotada para a convergência considerou a primeira época em que a variação percentual absoluta da perda média entre épocas consecutivas se manteve abaixo de 0,5%. A Figura 3 apresenta os resultados experimentais obtidos com as

diferentes configurações. A Figura 3(a) apresenta o comportamento da função de perda ao longo de 20 épocas para três tamanhos de lote: 10, 100 e 1000. O primeiro cenário, com lote igual a 10, resultou em uma redução modesta do erro de reconstrução, encerrando o treinamento com perda de 0,25. No segundo cenário, com lote 100, houve uma queda expressiva do erro nas 10 primeiras épocas, atingindo 0,15 ao final. No terceiro cenário, com lote 1000, observou-se o melhor desempenho, com erro final de 0,13, indicando maior eficiência na reconstrução dos dados de entrada. A Figura 3(b) avalia a influência do número de épocas na convergência da função de custo. Conforme o critério previamente definido (variação percentual $< 0{,}5\%$ entre épocas), o cenário com 20 épocas não converge, enquanto o cenário com lote 1000 e 50 épocas atinge convergência a partir da época 42. Essa observação motivou a escolha final da configuração da seleção de atributos. Essa observação, aliada ao menor tempo de execução por época evidenciado na Figura 3(c), na qual a configuração com lote 10 apresentou tempo de treino 21,2 vezes maior em comparação ao lote 1000, motivou a escolha do quarto cenário (lote 1000 e 50 épocas) como configuração final para a seleção de atributos. A partir dessa configuração, os atributos foram ordenados segundo seus respectivos erros de reconstrução, sendo selecionados os 75% com os menores valores para compor o conjunto final utilizado nos experimentos subsequentes.

Inicialmente, os modelos ResNet, LeNet-5 e FedPEAC-Net foram avaliados sem a aplicação de qualquer técnica de seleção de atributos. Na sequência, foi analisado o impacto da seleção de atributos sobre o desempenho de cada modelo. Os modelos ResNet e LeNet-5 foram adaptados a partir de suas versões tradicionais, de modo a permitir sua aplicação como classificadores em um sistema de detecção de intrusão federado. Como o experimento está inserido em um contexto de aprendizado federado, os resultados apresentados nos gráficos correspondem às médias ponderadas dos treinos e testes realizados pelos clientes. O treinamento federado foi conduzido ao longo de 20 rodadas, sendo que, em cada rodada, os clientes treinaram localmente por 20 épocas, com tamanho de lote de 4096 amostras. A seleção de atributos foi realizada com base nos dados de treinamento do cliente com o maior número de amostras. A Figura 4 apresenta os desempenhos dos três modelos antes da aplicação da seleção de atributos. As figuras na linha superior mostram os resultados nos dados de treinamento e as figuras na linha inferior, nos dados de teste.

Observando a Figura 4, nota-se que durante o treinamento os modelos apresentaram desempenho semelhante. Contudo, o modelo FedPEAC-Net, Figura 4(c), demonstrou convergência mais rápida. Nos dados de teste, o FedPEAC-Net superou os demais modelos em todas as métricas, atingindo $F_1 = 0,95$ na Figura 4(f), enquanto LeNet-5 e ResNet obtiveram $F_1 = 0,89$ e $F_1 = 0,78$. A ResNet apresentou sinais de sobreajuste, com uma queda acentuada de desempenho. Destaca-se, ainda, que o FedPEAC-Net apresentou curvas de perda mais suaves entre rodadas, sugerindo maior estabilidade.

A Figura 5 mostra que a seleção de atributos impactou negativamente todos os modelos. Embora seja um resultado esperado, uma vez que os atributos foram selecionados com base no erro de reconstrução, sem considerar relevância ou redundância para cada modelo específico, a quantificação do impacto negativo é importante para a avaliação da proposta. A ResNet manteve o padrão de sobreajuste, com $F_1 = 0,95$ no treino, mostrado na Figura 5(a), e $F_1 = 0,66$ no teste, Figura 5(d). O LeNet-5 apresentou $F_1 = 0,80$ no treino, Figura 5(b), e $F_1 = 0,76$ no teste, Figura 5(e). A proposta FedPEAC-Net manteve



(a) ResNet — treino. (b) LeNet-5 — treino. (c) FedPEAC-Net — treino.

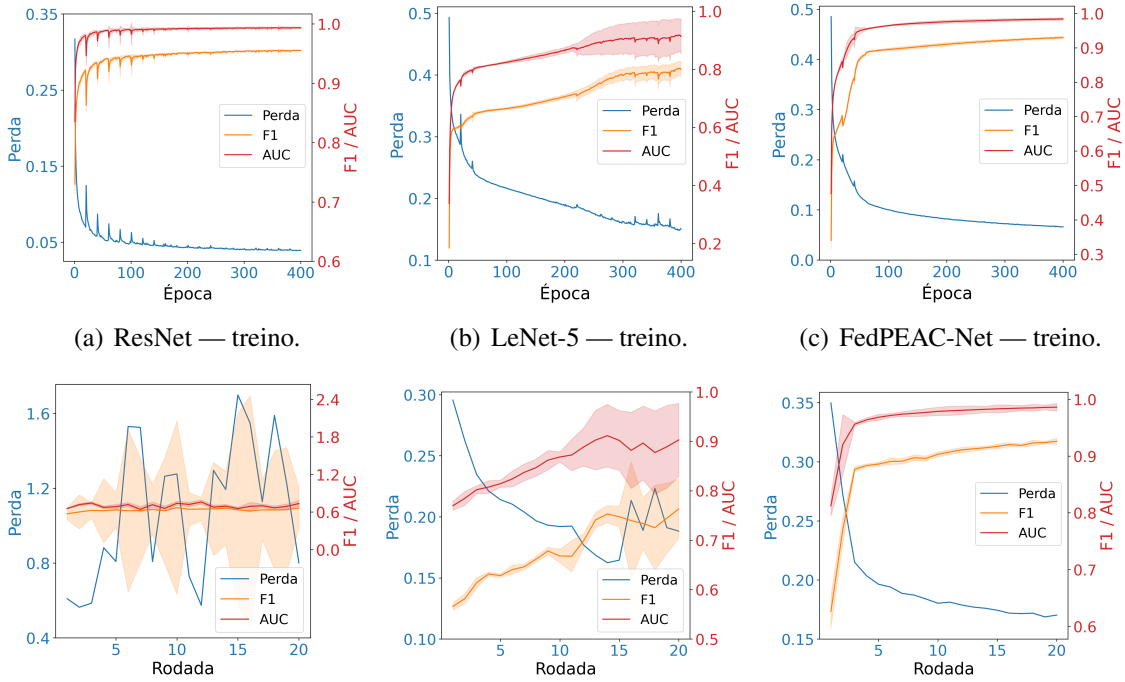
(d) ResNet — teste. (e) LeNet-5 — teste. (f) FedPEAC-Net — teste.

Figura 4. Desempenho dos modelos ResNet, LeNet-5 e FedPEAC-Net em ambiente federado antes da aplicação da seleção de atributos. Figuras superiores (a, b, c) referem-se aos dados de treinamento, enquanto as inferiores (d, e, f) mostram os resultados nos dados de teste.

desempenho mais estável, com $F_1 = 0,93$ tanto no treino, Figura 5(c), quanto no teste, Figura 5(f). Comparando os cenários antes e após a seleção de atributos, observam-se reduções no desempenho dos modelos na fase de teste. As quedas de F_1 foram de 15,6% para a ResNet, 15% para o LeNet-5 e apenas 2,3% para o FedPEAC-Net. Esses resultados indicam que a proposta de seleção de atributos apresenta maior compatibilidade com a arquitetura híbrida FedPEAC-Net, reforçando sua robustez e adaptabilidade em cenários federados. A métrica AUC-PR também foi considerada, com o modelo FedPEAC-Net obtendo 0,99 antes da seleção de atributos, na Figura 4(f), e 0,99, na Figura 5(f), após a seleção, mantendo o desempenho. Os modelos ResNet e LeNet-5 apresentaram reduções mais acentuadas, com AUC-PR caindo de 0,84, na Figura 4(d), para 0,74, na Figura 5(d), e de 0,97, na Figura 4(e), para 0,90, na Figura 5(e), respectivamente. Esses resultados corroboram a maior robustez do FedPEAC-Net diante da redução de dimensionalidade.

6. Conclusão

Este trabalho apresentou o modelo FedPEAC-Net, uma rede neural híbrida composta por um autocodificador e um classificador binário treinados em paralelo, juntamente com uma técnica de seleção de atributos baseada no erro de reconstrução. A proposta visa mitigar os impactos da alta dimensionalidade no desempenho de sistemas de detecção de intrusão em ambientes de aprendizado federado, especialmente com heterogeneidade como em redes IoT. Os resultados experimentais evidenciam a superioridade do FedPEAC-Net em relação a modelos clássicos como ResNet e LeNet-5, com destaque



(a) ResNet — treino. (b) LeNet-5 — treino. (c) FedPEAC-Net — treino.

(d) ResNet — teste. (e) LeNet-5 — teste. (f) FedPEAC-Net — teste.

Figura 5. Desempenho dos modelos ResNet, LeNet-5 e FedPEAC-Net após a aplicação da seleção de atributos. Gráficos superiores mostram os resultados no treinamento; os inferiores, na fase de teste.

para a F1 de 0,95 sem seleção de atributos e 0,93 após a seleção, com reduções significativamente menores que as observadas nos modelos tradicionais comparados. A combinação do treinamento paralelo com uma abordagem de seleção de atributos específica para a arquitetura do autocodificador contribuiu para um modelo mais robusto, generalizável e adequado a cenários distribuídos, demonstrando ser uma alternativa eficiente. Como trabalhos futuros, propõe-se uma seleção de atributos que pondere métricas de relevância e minimização do erro de reconstrução, sendo avaliado no modelo FedPEAC-Net.

Referências

Abdulganiyu, O. H., Ait Tchakoucht, T. e Saheed, Y. K. (2023). A systematic literature review for network intrusion detection system (IDS). *International Journal of Information Security*, 22(5):1125–1162.

Andreoni Lopez, M., Mattos, D. M. F., Duarte, O. C. M. B. e Pujolle, G. (2019). A fast unsupervised preprocessing method for network monitoring. *Annals of Telecommunications*, 74(3):139–155.

Barbosa, G. N. N., Andreoni, M. e Mattos, D. M. F. (2024). Optimizing feature selection in intrusion detection systems: Pareto dominance set approaches with mutual information and linear correlation. *Ad Hoc Networks*, 159:103485.

Bout, E., Loscri, V. e Gallais, A. (2022). How machine learning changes the nature of cyberattacks on iot networks: A survey. *IEEE Communications Surveys & Tutorials*, 24(1):248–279.

Cui, W., Lu, Q., Qureshi, A. M., Li, W. e Wu, K. (2020). An adaptive LeNet-5 model for anomaly detection. *Information Security Journal: A Global Perspective*, 30:19–29.

Cunha Neto, H. N., Hribar, J., Dusparic, I., Fernandes, N. C. e Mattos, D. M. (2024). Fedsbs: Federated-learning participant-selection method for intrusion detection systems. *Computer Networks*, 244:110351.

Cunha Neto, H. N., Mattos, D. M. F. e Fernandes, N. C. (2020). Privacidade do usuário em aprendizado colaborativo: Federated learning, da teoria à prática. *Minicursos do Simpósio Brasileiro de Segurança de Informação e de Sistemas Computacionais - SBSeg*, 20:142–195.

Hussain, F., Abbas, S. G., Husnain, M., Fayyaz, U. U., Shahzad, F. e Shah, G. A. (2020). IoT DoS and DDoS attack detection using ResNet. Em *2020 IEEE 23rd International Multitopic Conference (INMIC)*, p. 1–6.

Kasongo, S. M. e Sun, Y. (2020). Performance analysis of intrusion detection systems using a feature selection method on the UNSW-NB15 dataset. *Journal of Big Data*, 7(1).

Kunang, Y. N., Nurmaini, S., Stiawan, D., Zarkasi, A., Firdaus e Jasmir (2018). Automatic features extraction using autoencoder in intrusion detection system. Em *2018 International Conference on Electrical Engineering and Computer Science (ICECOS)*, p. 219–224.

Masters, D. e Luschi, C. (2018). Revisiting small batch training for deep neural networks.

Meng, Q., Catchpoole, D., Skillicom, D. e Kennedy, P. J. (2017). Relational autoencoder for feature extraction. Em *2017 International Joint Conference on Neural Networks (IJCNN)*, p. 364–371.

Neto, H. N. C., Hribar, J., Dusparic, I., Mattos, D. M. F. e Fernandes, N. C. (2023). A survey on securing federated learning: Analysis of applications, attacks, challenges, and trends. *IEEE Access*, 11:41928–41953.

Nimbalkar, P. e Kshirsagar, D. (2021). Feature selection for intrusion detection system in internet-of-things (IoT). *ICT Express*, 7(2):177–181.

Qin, Y. e Kondo, M. (2021). Federated learning-based network intrusion detection with a feature selection approach. Em *2021 International Conference on Electrical, Communication, and Computer Engineering (ICECCE)*, p. 1–6.

Sharafaldin, I., Habibi Lashkari, A. e Ghorbani, A. A. (2018). Toward generating a new intrusion detection dataset and intrusion traffic characterization. Em *Proceedings of the 4th International Conference on Information Systems Security and Privacy*. SCITEPRESS - Science and Technology Publications.

Silva, J. V. V., de Oliveira, N. R., Medeiros, D. S., Lopez, M. A. e Mattos, D. M. (2022). A statistical analysis of intrinsic bias of network security datasets for training machine learning mechanisms. *Annals of Telecommunications*, p. 1–17.

Wang, W., Liang, C., Chen, Q., Tang, L., Yanikomeroglu, H. e Liu, T. (2022). Distributed online anomaly detection for virtualized network slicing environment. *IEEE Transactions on Vehicular Technology*.