Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Botnet detection using federated learning with Energy-based Flow Classifier

Ândrey G. Mendes
João V. P. de Souza

Monografia apresentada como requisito parcial
para conclusão do Bacharelado em Ciência da Computação

Orientador
Prof. Dr. Marcelo Marotta

Brasília
2025

Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Botnet detection using federated learning with Energy-based Flow Classifier

Ândrey G. Mendes
João V. P. de Souza

Monografia apresentada como requisito parcial
para conclusão do Bacharelado em Ciência da Computação

Prof. Dr. Marcelo Marotta (Orientador)
CIC/UnB

Prof. Dr. João J. C. Gondim
ENE - DEPTO ENGENHARIA ELETRICA

Prof. Dr. Roberto Vito Rodrigues Filho
CIC/UNB

Prof. Dr. Marcelo Grandi Mandelli
Coordenador do Bacharelado em Ciência da Computação

Brasília, 14 de Julho de 2025

Dedicatória

Gostaríamos de dedicar este trabalho a todas as pessoas que conhecemos durante a graduação, aos grandes professores que nos ajudaram ao longo dessa trajetória e aos amigos que fizemos nessa jornada, que permanecem ao nosso lado até hoje. Mesmo diante das dificuldades encontradas no caminho, como a pandemia, sempre mantivemos o contato, o que contribuiu para o nosso progresso.

Agradecimentos

Agradecemos ao nosso orientador, professor Marcelo Marotta, por todo o apoio na elaboração deste trabalho, auxiliando-nos nas dificuldades encontradas ao longo do caminho. Também agradecemos ao nosso amigo Álvaro Veloso, por contribuir com nosso entendimento sobre o assunto durante o desenvolvimento do trabalho. Aos amigos que fizemos ao longo desta jornada na graduação, deixamos nossa gratidão. E, especialmente, à nossa família, por todo o suporte e incentivo durante esse período.

Resumo

Este trabalho investiga a aplicação do classificador baseado em fluxo de energia (EFC) no contexto de aprendizado federado para a detecção de tráfego malicioso em redes. Utilizam-se dados das bases CTU e ISOT, distribuídos entre dois e quatro dispositivos, com diferentes proporções entre treino e teste, simulando cenários realistas de heterogeneidade estatística e operacional. O estudo explora duas estratégias de agregação de modelos locais: média ponderada e média aritmética, com o objetivo de avaliar como o tipo de agregação e a distribuição dos pesos entre os dispositivos impactam o desempenho do sistema global. São empregadas métricas complementares, como F1-score e AUC-ROC, para medir a precisão, a robustez e a capacidade de generalização do classificador agregado. Os resultados indicam que a média ponderada tende a oferecer maior controle sobre a influência de dispositivos com qualidade superior de aprendizado, resultando em classificações mais eficazes, especialmente em cenários com alta variabilidade nas distribuições energéticas. Já a média aritmética, embora mais simples, mostra-se sensível à presença de modelos locais com desempenho inferior.

Palavras-chave: Energy-Based Flow Classifier, EFC, Botnet,Aprendizagem Federada

Abstract

This work investigates the use of the Energy Flow Classifier (EFC) in a federated learning setting for malicious traffic detection in computer networks. The experiments leverage data from the CTU and ISOT datasets, distributed across two and four devices to simulate realistic conditions of statistical and operational heterogeneity. Both weighted and unweighted (arithmetic mean) aggregation strategies are explored to evaluate how the combination of local models affects overall system performance. The study employs complementary metrics, such as F1-score and AUC-ROC, to assess the accuracy, robustness, and generalization capacity of the aggregated classifier. Results show that weighted aggregation offers greater control over the influence of high-quality local models, leading to more effective detection, especially in environments with high energy distribution variability. In contrast, the arithmetic mean is more sensitive to underperforming models, which can reduce classification reliability in heterogeneous scenarios.

Keywords: Energy-Based Flow Classifier, EFC, Botnet, Federated Learning

Sumário

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

A crescente sofisticação das botnets, redes de dispositivos comprometidos usados para realizar atividades maliciosas, representa uma ameaça significativa para a segurança cibernética, [1]. Essas botnets podem ser empregadas para conduzir uma variedade de ataques, como DDoS, roubo de informações, distribuição de malware, entre outros tipos de cibercrime [2]. A complexidade de mitigar essas ameaças aumenta devido a distribuição geográfica e a diversidade de redes dos dispositivos comprometidos [3][4]. Essa dispersão em diferentes locais dificulta a identificação centralizada de botnets, pois os dispositivos podem estar espalhados por diferentes regiões do mundo. Assim a necessidade de estratégias mais eficazes e descentralizadas de detecção torna-se mais evidente [5].

As redes distribuídas e a diversidade de locais geográficos complicam ainda mais a detecção de botnets, que tradicionalmente dependem de análise centralizadas [3] [5]. Essas abordagens envolvem a coleta de processamento de grandes volumes de dados de rede em servidores centrais para a identificação de padrões de comportamentos suspeitos . No entanto, essa abordagem enfrenta desafios consideráveis, como a exigência de grande largura de banda para a transferência de dados [6][4]. Além disso, a dificuldade de lidar com dados heterogêneos provenientes de diferentes ambientes torna a detecção centralizada menos eficaz [2][7]. A necessidade de soluções inovadoras para superar essas limitações é crucial para a evolução da cibersegurança [5][4].

O aprendizado federado surge como uma solução promissora para superar as limitações das abordagens centralizadas [8][5][3]. Essa técnica de aprendizagem de máquina permite o treinamento de modelos de forma descentralizada, eliminando a necessidade de transferir dados brutos para um servidor central. Em vez disso, cada dispositivo ou nodo de rede treina localmente um modelo utilizando seus próprios dados, o que preserva a privacidade e reduz a necessidade de largura de banda [8] [7]. Apenas as atualizações do modelo,

como gradientes, são compartilhados com um servidor central. Este servidor central então agrega essas atualizações para construir um modelo global robusto e eficaz na deteção de ameaças [8] [4].

No estudo de Pontes,*et. al* [9] foi proposto um classificador chamado *Energy-based Flow Classifier (EFC)*, que utiliza estatísticas inversas para detectar anomalias. Embora essa técnica tenha sido concebida para uma ampla gama de aplicações, incluindo a detecção de anomalias em geral, ela também foi aplicada com sucesso na detecção de botnets, como demonstrado na pesquisa de Lopes, Daniele A. G. *et al.* [10], onde foi realizada uma análise exploratória dos mecanismos de detecção de botnets com base no comportamento do fluxo de rede, utilizando o *EFC* como técnica principal.

Neste trabalho, propomos um detector de botnets baseado em aprendizado federado utilizando o algoritmo *Energy-based Flow Classifier (EFC)*. O objetivo central é investigar os efeitos da integração do aprendizado federado sobre o desempenho do EFC na tarefa de detecção de botnets. A proposta busca avaliar se a agregação descentralizada de modelos, característica do aprendizado federado, compromete ou potencializa a eficácia do EFC, que originalmente opera com dados centralizados. Inspirado no modelo de Potts inverso da mecânica quântica e adaptado para a classificação de fluxos de rede, o *EFC* utiliza apenas dados benignos, sem a necessidade de aprender os comportamentos dos dados maliciosos para a análise de anomalias, tornando-se adaptável a diferentes domínios [10] [9]. Com essas características, o *EFC* mostra-se promissor para a detecção de botnets. Para avaliar a eficiência do modelo, propomos a integração do aprendizado federado, que permitirá a detecção de botnets de forma descentralizada e robusta, garantindo maior flexibilidade e precisão na identificação de novas ameaças, independentemente da estrutura ou dos protocolos utilizados [5][3][7][4].

Capítulo 2

Fundamentação teórica

Este capítulo apresenta os fundamentos teóricos que sustentam o desenvolvimento deste trabalho, com foco em dois pilares principais: o classificador baseado no fluxo de energia ($EnergyFlowClassifier - EFC$) e a aprendizagem federada. Inicialmente, discute-se o conceito do classificador EFC. Em seguida, é introduzida a aprendizagem federada como uma abordagem distribuída de treinamento de modelos de aprendizado de máquina, na qual os dados permanecem localmente em dispositivos ou nós participantes, reduzindo a necessidade de transferência massiva de dados. A compreensão desses conceitos é essencial para o entendimento da proposta metodológica deste trabalho, que busca integrar as vantagens da aprendizagem federada com a capacidade discriminativa do EFC em cenários distribuídos de análise de dados.

2.1 Energy-based Flow Classifier

O Energy-based Flow Classifier (EFC), proposto por Pontes, é um classificador de fluxo de energia baseado na estatística inversa do modelo de Potts[9]. Seu objetivo é inferir características de uma estatística global a partir de sua distribuição amostral. Esse modelo representa spins interagindo em uma liga cristalina, contendo uma descrição matemática. No modelo de Potts, adaptado para fluxos de rede, cada fluxo k é representado por aspecto específico do gráfico $G_k(\eta, \varepsilon)$ podendo ser visto na figura 2.1. Em cada fluxo k, os nós correspondem às características i, onde $i \in \eta$, que representam as propriedades do fluxo, como $DstPort$, $Protocol$, $FlowDuration$, entre outras. Para cada i, é atribuído um valor a_{ki}, pertencente ao conjunto Ω_i, o qual contém todos os valores possíveis daquele atributo específico, a cada valor a_{ki} é associado um campo local $h_i(a_{ki})$.

Além disso, o conjunto de arestas do grafo G_k, composto por todos os pares possíveis de atributos, é representado por $\varepsilon = \{(i,j)|i,j \in \eta; i \neq j\}$. Cada aresta possui um valor de acoplamento associado, definido pela função $e_{ij}(a_{ki}, a_{kj})$.

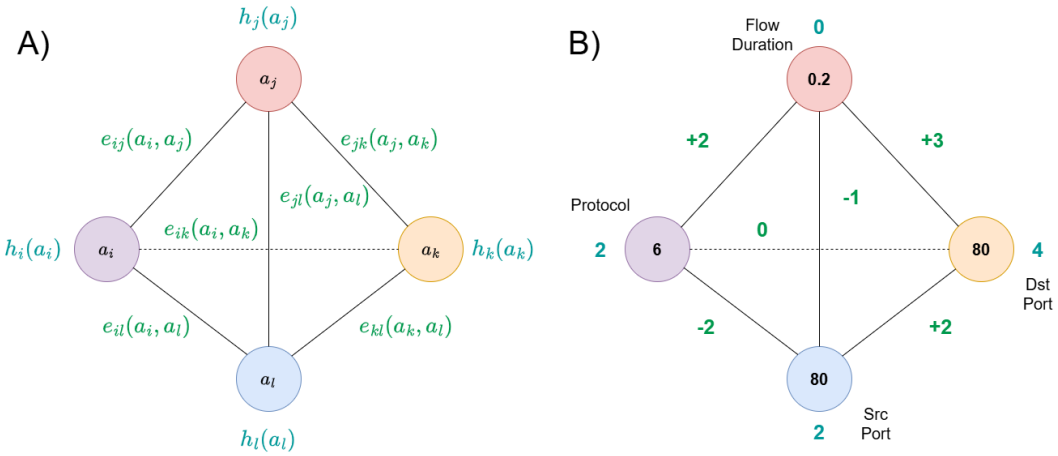

Figura 2.1: A) Interação dos spins no modelo Potts. B) Modelo adaptado para um fluxo de redes mapeado em um grafo.

Com os valores dos campos locais e de acoplamento, é possível obter a "energia"total de cada fluxo k, determinado pelo Hamiltoniano $\mathscr{H}(a_{k1} \ldots a_{kN})$, análogo à noção de Hamiltoniano na Mecânica Quântica. Um ponto a ser destacado é que o modelo descrito é discreto, assim sendo, os atributos contínuos devem ser discretizados para se adaptar ao modelo

2.1.1 Modelo de inferência

A ideia é obter um modelo estatístico a parti de amostra de fluxos, e inferir valores de acoplamento e campos locais que caracterizam esse tráfego específico. Sendo calculado a energia do fluxo por meio dos parâmetros inferidos, esperando que os fluxos benignos tenham valores menores de energias do que os maliciosos. Para inferir um modelo estatístico, Pontes usa estatística inversa, nela a probabilidade $P(a_{k1} \ldots a_{aN})$ é associada a cada fluxo $k \in \mathcal{K}$, com base no conjunto $\mathcal{S}$. Sendo $\mathcal{K}$ o conjunto de todos os fluxos possíveis, significando todas as possíveis combinações de valores ($\mathcal{K} = \Omega^{\mathrm{N}}$), e $\mathcal{S}$ um conjunto de fluxos ($\mathcal{S} \subset \mathcal{K}$). Com isso, o modelo global P é inferido a parti do Princípio da Maximização da Entropia:

$$\max_{P} \; -\sum_{k\in\mathcal{K}} P(a_{k1}\ldots a_{k\mathrm{N}})\log\left(P(a_{k1}\ldots a_{k\mathrm{N}})\right) \tag{2.1}$$

$$\sum_{k\in\mathcal{K}|a_{ki}=a_i} P(a_{k1}\ldots a_{k\mathrm{N}}) = f_i(a_i) \qquad \forall i\in\eta;\,\forall a_i\in\Omega; \tag{2.2}$$

$$\sum_{k\in\mathcal{K}|a_{ki}=a_i} P(a_{k1}\ldots a_{k\mathrm{N}}) = f_{ij}(a_i,a_j) \qquad \forall (i,j)\in\eta^2 \mid i\neq j;\,\forall (a_i,a_j)\in\Omega^2; \tag{2.3}$$

Para garantir que o modelo reproduza os dados empíricos, são geradas as frequências empíricas simples $f_i(a_i)$ e duplas $f_{ij}(a_i,a_j)$ como marginais, conforme apresentado nas restrições 2.2 e 2.3, correspondendo, respectivamente, à frequência empírica de a_i e de (a_i,a_j). Ambas as frequências são obtidas a partir do conjunto $\mathcal{S}$, por meio da contagem de ocorrências de um dado valor de a_i ou de pares de valores de atributos (a_i,a_j), divididas pelo número total de fluxos em $\mathcal{S}$. Entretanto, como o conjunto $\mathcal{S}$ é menor que o conjunto $\mathcal{K}$, uma vez que $\mathcal{S}\subset\mathcal{K}$, a inferência baseada em $\mathcal{S}$ está sujeita a efeitos de subamostragem (*undersampling*). Para mitigar esse efeito, utiliza-se pseudocontagem na estimativa das frequências empíricas, resultando nas correções:

$$f_i(a_i) \leftarrow (1-\alpha)f_i(a_i) + \frac{\alpha}{Q} \tag{2.4}$$

$$f_{ij}(a_i,a_j) \leftarrow (1-\alpha)f_{ij}(a_i,a_j) + \frac{\alpha}{Q^2} \tag{2.5}$$

Onde $(a_i,a_j)\in\Omega^2$ e $0\leq\alpha\leq 1$ é um parâmetro configurável do EFC, que define o peso relativo das pseudocontagens. A introdução da pseudocontagem é uma forma de assumir que o conjunto $\mathcal{S}$ seja estendido para incluir uma fração adicional de fluxos com valores de atributos distribuídos uniformemente. A maximização proposta pode ser resolvida utilizando a distribuição do tipo Boltzmann-like 2.6. No qual o objetivo é calcular o energia de cada fluxo individual, descrito em 2.7.

$$P^*(a_{k1}\ldots a_{k\mathrm{N}}) = e^{-\mathscr{H}(a_{k1}\ldots a_{k\mathrm{N}})} \tag{2.6}$$

$$\mathscr{H}(a_{k1}\ldots a_{k\mathrm{N}}) = -\sum_{i,j|i<j} e_{ij}(a_{ki},a_{kj}) - \sum_i h_i(a_{ki}) \tag{2.7}$$

Conforme descrito em 2.7, o Hamiltoniano é determinado em termos dos multiplicadores de Lagrange e_{ij} e h_i, associados às restrições 2.2 e 2.3, respectivamente. Nas condições do modelo de Potts, $e_{ij}(a_i, a_j) \mid (a_i, a_j) \in \Omega^2$ representa o conjunto de todos os possíveis valores de acoplamento entre dois atributos i e j, enquanto $h_i(a_{ki}) \mid a_i \in \Omega$ representa o conjunto de todos os possíveis campos locais associados a um atributo i.

A inferência dos acoplamentos $e_{ij}(a_i, a_j)$ é realizada por meio de uma aproximação Gaussiana, utilizando a inversão da matriz de covariância, conforme descrito em:

$$\begin{aligned} e_{ij}(a_i, a_j) &= -(C^{-1})_{ij}(ai, aj), \\ \forall (i,j) \in \eta^2, \forall (a_i, a_j) \in \Omega^2, a_i, a_j &\neq Q \end{aligned} \tag{2.8}$$

onde

$$C_{ij}(a_i, a_j) = f_{ij}(a_i, a_j) - f_i(a_i) f_j(a_j) \tag{2.9}$$

é a matriz de covariância adquirida a partir das frequências empíricas simples e conjuntas, e a inversão da matriz de covariância é feita para remoção dos efeitos de correlações indiretas nos dados. Um adendo importante é que a inferência dos campos locais e dos acoplamentos possui um número de parâmetros maior que o número de restrições independentes. Consequentemente, para eliminar esse problema, sem perda de generalidade, define-se:

$$e_{ij}(a_i, Q) = e_{ij}(Q, a_i j) = h_i(Q) = 0 \tag{2.10}$$

portanto não precisa calcular $e_{ij}(a_i, a_j)$ caso o a_i ou a_j seja igual a Q.

Para a inferência dos campos locais $h_i(a_i)$, foi utilizada a aproximação de campos médios, na qual a interação de um atributo com seus vizinhos é substituída pela interação aproximada com a média dos atributos. O objetivo é obter um valor aproximado para os campos locais associados, sendo realizado da seguinte forma:

$$\begin{aligned} \frac{f_i(a_i)}{fi(Q)} = \exp\left(h_i(a_i) + \sum_{j, a_j} e_{ij}(a_i, a_j) f_j(a_j) \right), \\ \forall i, \in \eta, a_i, \in \Omega, a_i \neq Q \end{aligned} \tag{2.11}$$

onde $f_i(Q)$ é a frequência do último elemento $a_i = Q$, para qualquer atributo i, utilizada para essa normalização. É importante destacar que o elemento Q foi selecionado arbitrariamente, podendo ser qualquer valor de $\Omega = 1 \ldots Q$, desde que o valor escolhido seja mantido para o cálculo dos campos locais em todos os atributos $i \in \eta$. Portanto, com os

valores conhecidos, os campos locais podem ser calculados com as frequências empíricas simples $f_i(a_i)$ e os acoplamentos $e_{ij}(a_i, a_j)$ da seguinte forma:

$$h_i(a_i) = \ln\left(\frac{f_i(a_i)}{f_i(Q)}\right) + \sum_{j,a_j} e_{ij}(a_i, a_j) f_j(a_j) \tag{2.12}$$

A energia de um fluxo pode ser calculada a partir da equação 2.7, usando como referência os valores dos parâmetros e atributos do fluxo do modelo inferido na seção acima. O cálculo consiste em uma soma negativa dos acoplamentos e dos campos locais. Caso um fluxo contenha atributos semelhantes aos dos fluxos inferidos, apresentará valores de energia mais baixos. Com base nisso, é possível calcular o limiar de energia para a classificação das classes apresentadas, permitindo identificar se um determinado fluxo pertence a uma classe quando sua energia fica abaixo do limiar, ou se não pertence, ao apresentar uma energia superior, de acordo com a inferência do modelo.

2.2 Aprendizagem Federada

O Aprendizado Federado (FL) funciona por meio de uma arquitetura colaborativa entre múltiplos participantes, como dispositivos ou instituições, que treinam um modelo de forma descentralizada [8]. O processo começa com a distribuição de um modelo global inicial a todos os participantes a partir de um servidor central. Cada participante então realiza o treinamento local desse modelo usando seus próprios dados, sem compartilhar essas informações com o servidor. Após esse treinamento, os participantes enviam de volta apenas os parâmetros atualizados do modelo. O servidor central recebe essas atualizações e as agrega para formar uma nova versão do modelo global. A figura 2.2 representa uma arquitetura generalizada do modelo de aprendizagem federada.

O mecanismo de agregação dos parâmetros é um componente fundamental, sendo o algoritmo FedAvg um dos mais utilizados. Nesse processo, o servidor calcula uma média ponderada das atualizações enviadas pelos participantes, levando em conta o tamanho dos dados locais [8]. Essa nova versão agregada do modelo é então redistribuída para todos os participantes para a próxima rodada. Com cada ciclo, o modelo global vai se tornando mais representativo e preciso, incorporando gradualmente o conhecimento extraído dos dados locais. O objetivo é alcançar uma convergência eficiente, mesmo com a diversidade dos dados distribuídos entre os clientes.

O aprendizado federado pode ser estruturado em diferentes configurações de participação, como cross-silo ou cross-device [8][5][3]. No cross-silo, os participantes são instituições ou organizações com infraestrutura mais estável, como hospitais, bancos ou provedores

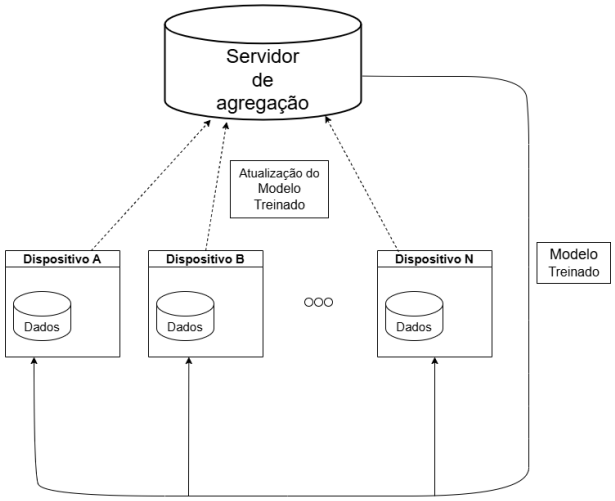

Figura 2.2: Representação do modelo federado.

de internet. Já no cross-device, a aprendizagem ocorre entre dispositivos finais, como smartphones e sensores IoT, geralmente em grande escala e com maior variabilidade de conexão e poder computacional [5][2][7]. A escolha da configuração afeta diretamente as estratégias de sincronização, comunicação e robustez necessárias ao sistema. Ambos os modos compartilham o mesmo princípio de manter os dados locais, mas exigem abordagens técnicas distintas. Isso permite aplicar esse modelo em diferentes cenários, desde ambientes corporativos até redes distribuídas de dispositivos pessoais [3][7].

Outro aspecto importante do seu funcionamento é a sincronização e a seleção dos participantes. Em cada rodada, nem todos os participantes precisam estar ativos, podendo selecionar apenas um subconjunto com base em critérios como disponibilidade, conectividade ou qualidade dos dados [2][7]. Essa seleção reduz a sobrecarga de comunicação e melhora a escalabilidade do sistema. Além disso, pode operar de forma síncrona, onde todos os participantes aguardam a conclusão da rodada, ou de forma assíncrona, em que as atualizações são incorporadas à medida que chegam[6][4]. Essa flexibilidade contribui para a eficiência e adaptabilidade do processo.

Dessa forma, o aprendizado federado oferece uma solução inovadora para treinar modelos de aprendizado de máquina sem centralizar os dados. A principal vantagem está na preservação da privacidade, já que os dados permanecem nos dispositivos dos usuários ou silos institucionais. Apenas os parâmetros do modelo são compartilhados com o servidor central, reduzindo os riscos de exposição de informações sensíveis [8][5][3]. Com isso, é possível cumprir requisitos legais de proteção de dados, como o GDPR e a LGPD, mantendo a utilidade dos dados para aprendizado.

Outro benefício é a redução do tráfego de rede e dos custos com armazenamento centralizado. Em vez de transferir grandes volumes de dados para um servidor, os dispositivos apenas enviam atualizações de parâmetros, que são geralmente muito menores [4][6]. Isso é particularmente útil em aplicações com dispositivos de borda, como sensores IoT e smartphones, que geram dados continuamente [2]. A economia de largura de banda também contribui para maior eficiência energética e menos sobrecarga de rede, viabilizando aplicações em ambientes com conectividade limitada[5].

Esse modelo também promove a colaboração entre múltiplas entidades que, de outra forma, não poderiam compartilhar dados diretamente. Instituições concorrentes ou com políticas rígidas de privacidade podem colaborar para treinar um modelo conjunto sem expor seus ativos informacionais [11][6]. Isso estimula a criação de modelos mais robustos, baseados em uma diversidade maior de dados. Quanto maior a variedade de dados, melhor a capacidade de generalização do modelo treinado. Essa colaboração é particularmente vantajosa em áreas como detecção de ameaças cibernéticas, onde diferentes organizações enfrentam tipos variados de ataques. Com a aprendizagem federada, é possível unir forças sem comprometer a segurança dos dados.

Além disso, permite atualizações frequentes e personalizadas do modelo, refletindo a realidade local de cada participante. Isso significa que os modelos podem aprender continuamente com dados novos e específicos de cada ambiente, adaptando-se rapidamente a mudanças no padrão de uso ou comportamento. Em sistemas como assistentes virtuais ou detectores de intrusão em rede, essa personalização local pode melhorar consideravelmente a performance. Ao mesmo tempo, a atualização global agrega conhecimento útil de múltiplas fontes. Isso resulta em modelos mais eficientes, adaptáveis e centrados no usuário.

Por tanto, oferece resiliência e robustez ao sistema, já que não depende de um único ponto de falha. Em modelos tradicionais, a interrupção no servidor central ou a perda de dados pode comprometer todo o processo. Já no modelo federado, mesmo que alguns participantes falhem ou se desconectem, o treinamento pode continuar com os demais. Essa característica o torna ideal para sistemas distribuídos e ambientes críticos. A descentralização também reduz os riscos de ataques direcionados ao armazenamento central, aumentando a segurança do sistema como um todo. Assim, modelo federado não apenas preserva dados, mas também fortalece a arquitetura de aprendizado.

Apesar disso, a aprendizagem federada também enfrenta desafios significativos. Um dos principais obstáculos é a heterogeneidade dos dados entre os participantes. Em muitos casos, os dados locais seguem distribuições diferentes, o que compromete a convergência e a qualidade do modelo global [8][2]. Esse cenário é conhecido como non-IID e pode gerar

resultados enviesados, já que cada participante contribui com um modelo adaptado à sua realidade local. Isso exige estratégias de agregação mais sofisticadas para compensar a variabilidade. Modelos como FedProx, FedAvgM e FedOpt têm sido propostos justamente para lidar com esse tipo de complexidade [4].

As limitações de recursos computacionais e conectividade também impactam negativamente o desempenho, especialmente em ambientes com dispositivos de borda como sensores, smartphones e gateways IoT [5][2][11]. Esses dispositivos geralmente têm pouca capacidade de processamento, memória e bateria, o que dificulta a execução de modelos complexos. Além disso, conexões instáveis podem levar à perda de atualizações, atrasos ou até exclusão de participantes em determinadas rodadas. Isso compromete a continuidade e a consistência do processo federado. Para contornar isso, é comum aplicar estratégias de seleção de clientes, participação assíncrona ou adaptação da complexidade do modelo.

A agregação dos modelos locais é outro ponto técnico crucial. Estratégias simples como FedAvg funcionam bem apenas quando os dados são homogêneos e as condições dos participantes são similares. No entanto, em cenários reais, com dados diversos e infraestrutura desigual, essas abordagens perdem eficiência e precisão. Por isso, novos algoritmos têm sido propostos para levar em conta a variabilidade nas contribuições locais, o tamanho dos dados, o número de épocas de treinamento e outros fatores [6][4]. Mesmo assim, a robustez dessas soluções ainda está em estudo, principalmente para garantir estabilidade em grandes escalas.

A escalabilidade também é um desafio importante, especialmente em cenários do tipo cross-device, com milhares ou milhões de dispositivos participantes [5][6]. Coordenar a participação, garantir conectividade mínima e manter o desempenho do modelo são tarefas difíceis quando o número de clientes cresce significativamente. Além disso, há um aumento expressivo no custo de comunicação e no tempo de treinamento por rodada. Estratégias como amostragem de clientes, comunicação comprimida e treinamento assíncrono têm sido utilizadas para contornar essas limitações [7][6][4]. Mesmo assim, a escalabilidade continua sendo um dos principais gargalos para aplicações práticas em larga escala.

A avaliação do modelo global em ambientes federados também apresenta dificuldades. Como os dados permanecem distribuídos e não são acessíveis diretamente pelo servidor, medir métricas de desempenho de forma precisa é um desafio. Além disso, o desempenho pode variar entre diferentes grupos de usuários, o que exige avaliações mais granulares e representativas. Estratégias como avaliação local dos participantes ou validação cruzada entre silos têm sido usadas, mas ainda não há um padrão estabelecido. Essa limitação afeta a confiança e a interpretação dos resultados obtidos com FL.

2.3 Base de dados

A escolha das Base de dados utilizados para os experimentos desse trabalho, baseia-se no de Lopes, Daniele A. G. et al. [10]. Esses dois conjuntos de dados (CTU-13, ISOT-HTTP) foram selecionados por serem amplamente utilizados na literatura, devido à sua relevância e realismo. Ambos contêm fluxos de rede com diferentes tipos de ataques de botnets. Além disso, foi necessário utilizar uma base de dados contendo apenas fluxos benignos para complementar o CTU-13, uma vez que os tráfegos benignos não estão disponíveis publicamente."

Os conjuntos de dados foram gerados no formato de captura de pacotes (PCAP) e, para a extração dos fluxos de rede dos tráfegos capturados, foi utilizada a ferramenta CICFlow-Meter. O CICFlowMeter é um gerador e analisador de fluxos de tráfego de rede fornecido pelo Canadian Institute for Cybersecurity [12]. Nele, os fluxos são gerados de forma bi-direcional, sendo que o primeiro pacote define as direções de ida (origem para destino) e volta (destino para origem). Além disso, fluxos TCP são encerrados geralmente ao final da conexão (por meio do pacote FIN), enquanto fluxos UDP são finalizados com base em um tempo limite (timeout), que pode ser definido arbitrariamente, como por exemplo, 600 segundos para ambos os protocolos TCP e UDP. Após a extração, é gerado um arquivo CSV contendo 84 colunas com estatísticas de tráfego. Em seguida, serão detalhadas as suas características.

2.3.1 CTU-13

O CTU-13 [1] é um conjunto de dados de tráfego de botnets capturado na Universidade CTU, na República Tcheca, em 2011. Seu objetivo era registrar uma grande quantidade de tráfego real de botnets, mesclado com tráfego normal e de background. O conjunto é composto por treze capturas (denominadas cenários), cada uma representando amostras diferentes de botnets. Cada cenário executa um malware específico, envolvendo diversos protocolos e realizando distintas ações.

Para os experimentos, foram selecionados os cenários 1, 3, 5, 6, 7, 8 e 12. Consequentemente, estão representados sete tipos de botnets: Neris, Rbot, Virut, Menti, Sogou, Murlo e Nsis.ay. A Tabela 2.1 apresenta suas principais características. A combinação das botnets inclui estruturas centralizadas e descentralizadas. Devido a preocupações com a privacidade, os arquivos PCAP disponibilizados contêm apenas o tráfego malicioso, não estando disponível publicamente a captura completa, que inclui tráfego de *background*, botnets e tráfego normal. Assim, este trabalho utilizou os dados maliciosos disponíveis publicamente para conduzir as análises.

Tabela 2.1: Características dos cenários de botnets

Cenário	IRC	SPAM	CF	PS	US	P2P	HTTP
1	✓	✓	✓				
3	✓			✓	✓		
5		✓		✓			✓
6				✓			
7							✓
8				✓			
12						✓	

IRC: Internet Relay Chat
CF: ClickFraud.
PS: Port Scan.
US: Compilado e controlado pelos autores.
P2P: Peer to Peer
HTTP: Hypertext Transfer Protocol

2.3.2 CICIDS2017

Para obter fluxos de dados benignos pros testes com o CTU-13, foi escolhido a base de dados CICIDS2017 [13], do Canadian Institute for Cybersecurity . Utilizado no trabalho anteriores com o EFC, o conjunto contém dados benignos e maliciosos atualizados. Neste trabalho, foi selecionada a captura referente ao dia 3 de julho de 2017 (segunda-feira), que representa um dia normal e inclui apenas tráfego benigno. O arquivo CSV correspondente possui 79 colunas.

2.3.3 ISOT HTTP

O ISOT HTTP [14] foi desenvolvido pela Universidade de Victoria, no Canadá. Ele consiste em dois tipos de conjuntos de dados: o primeiro é composto por tráfego DNS malicioso gerado por diferentes botnets; o segundo, por tráfego DNS benigno, gerado por diferentes aplicativos de software conhecidos. Todos utilizam protocolos HTTP, com os dados maliciosos gerando cinco arquivos PCAP, e os benignos três aquivos PCAP.

Os dados maliciosos foram coletados em um ambiente virtual com nove tipos de exploit kits. Cada bot foi implantado em uma máquina virtual com Windows XP, configurada para se comunicar com um servidor específico de comando e controle (C&C), resultando em nove tipos de servidores, um para cada botnet. Todos utilizam arquitetura centralizada. Os dados benignos foram coletados a partir de aplicações legítimas em um ambiente virtual. Cada software foi instalado em uma máquina virtual com Windows 7, incluindo exemplos como Skype, Avast e Adobe.

Para o experimento, foram utilizados os três arquivos de tráfego benigno e o arquivo init4.pcap como malicioso. Este último contém todos os tipos de botnets presentes no ISOT HTTP, sendo eles: Zyklon, Bluebot, Liphyra, Gaudox, Blackout, Citadel, Be.botnet e Zeus. Os arquivos continham 84 colunas onde pode ser visualizado na tabela 2.2. A característica *Unknown* deve-se ao fato de o dataset conter aplicações que não correspondem nem às aplicações benignas nem aos servidores dos fluxos maliciosos, sendo, portanto, alocadas como desconhecidas.

Tabela 2.2: Amostra do ISOT HTTP

Label	**Quantidade**
Benign	256.348
Citadel	145.088
Gaudox	90.972
Zeus	80.654
Unknown	14.630
Be.botnet	13.755
Bluebot	13.593
Zyklon	12.008
Blackout	6.881
Liphyra	3.782
Total Maliciosos	366.733
Total Benigno	256.348
Total Unknown	14.630

2.4 Resumo

A fundamentação teórica do Energy Flow Classifier (EFC) e do aprendizado federado apresenta importância para o entendimento de seus conceitos mais relevantes. Com o estudo do EFC, foi possível compreender melhor os valores de acoplamento e os campos locais, além da ideia de inferi-los para o cálculo da energia. No caso do aprendizado federado, foi importante entender seus conceitos, especialmente o uso descentralizado, em que apenas os parâmetros do modelo são enviados ao servidor, que os atualiza por meio de um algoritmo de agregação com o objetivo de gerar uma versão mais robusta do modelo. Com isso, compreendeu-se que é possível utilizar ambos no desenvolvimento deste trabalho, empregando os valores de acoplamento e os campos locais como parâmetros do modelo a serem agregados em um ambiente federado. Além disso, foi possível aprofundar o entendimento sobre os conjuntos de dados utilizados, reconhecendo os tipos de botnets e os dados benignos oferecidos, bem como suas características.

Capítulo 3

Trabalhos Relacionados

Neste capítulo, serão discutidos os trabalhos mais recentes da literatura relacionados à detecção de intrusão em fluxos de rede, com foco em botnets utilizando aprendizado federado. Serão abordadas as diferentes abordagens utilizadas, destacando suas vantagens em relação aos métodos de aprendizado centralizado. Além disso, são apresentados trabalhos que utilizam o EFC, evidenciando sua flexibilidade e desempenho na detecção de intrusões. Encerrando com uma discussão sobre eles e sobre como se relacionam com este trabalho.

3.1 Federated Deep Learning for Zero-Day Botnet Attack Detection in IoT-Edge Devices

Nos últimos anos, várias propostas foram exploradas para a detecção de botnets utilizando aprendizado federado. Um exemplo disso é apresentado no trabalho de Popoola *et al* [5], onde se propõe o uso de aprendizado profundo federado (FDL) para detectar ataques de botnets zero-day. Nesse método, uma arquitetura de rede neural profunda (DNN) é utilizada para a classificação do tráfego de rede. O uso de aprendizado federado permite uma abordagem mais segura e distribuída, sem a necessidade de compartilhar dados sensíveis entre dispositivos.

O modelo de DNN é treinado localmente de forma independente em vários dispositivos de borda IoT, enquanto o algoritmo de média federada (FedAvg) é empregado para agregar as atualizações dos modelos locais. Após várias rodadas de comunicação entre o servidor de parâmetros e os dispositivos de borda IoT, é gerado um modelo global de DNN. Esse processo garante que os dados permaneçam localmente, melhorando a eficiência do treinamento distribuído.

Como comparação ao FDL, foram analisados outros algoritmos, como o aprendizado profundo centralizado (CDL), localizado (LDL) e distribuído (DDL). Os resultados mostraram que os modelos CDL e FDL obtiveram melhor desempenho na classificação em comparação com os modelos LDL e DDL. No entanto, o FDL apresentou vantagens, como menor sobrecarga de comunicação, menor demanda de espaço de memória para armazenamento de dados e baixa latência em relação ao CDL. Em contrapartida, uma desvantagem observada foi o tempo relativamente alto necessário para o treinamento do modelo, quando comparado aos outros algoritmos.

A análise deste estudo revela diversas ligações para o nosso projeto. Em primeiro lugar, ele apresenta uma estrutura sólida de desenvolvimento, além de oferecer metodos que podem acelerar o progresso do nosso sistema. O algoritmo de agregação FedAvg, utilizado na pesquisa, mostrou-se bastante eficaz, representando uma técnica promissora para combinarmos os modelos treinados localmente de forma eficiente. Com isso, extraímos os pontos mais relevantes da proposta, adaptando-os às necessidades e particularidades do nosso contexto.

3.2 Towards Detection of Zero-Day Botnet Attack in IoT Networks using Federated Learning

Outra pesquisa relevante é o estudo de Zhang *et al* [3], que propõe uma estrutura baseada em aprendizado federado (FL) para treinar modelos voltados à detecção de ataques de botnets zero-day. O algoritmo de agregação K-greedy foi desenvolvido com o objetivo de permitir uma detecção mais eficiente desses ataques em dispositivos, sem a necessidade de registros prévios dos ataques, o segundo é a necessidade de uma agregação eficiente de dados, considerando a escala das redes IoT. Resolvendo a necessidade de uma agregação eficiente de dados, considerando a escala das redes IoT.

Diferente de métodos como o FedAvg, que não considera a incerteza nos dados, o K-greedy foca nos dispositivos com maior incerteza de detecção. Ele utiliza modelos estocásticos de aprendizado bayesiano profundo para medir essa incerteza. A avaliação é feita por meio de medidas da teoria da informação. Esse processo otimiza a detecção de ataques zero-day.

Os resultados mostram que o K-greedy superou significativamente os algoritmos FedSGD e FedAvg na detecção de ataques botnet em um ambiente FL. Utilizando o conjunto de dados NBaIoT, que inclui tráfego de dispositivos IoT infectados por botnets como Mirai e BASHLITE, o K-greedy alcançou uma precisão de 90% para detecção de ataques zero-day.

Mesmo dispositivos sem registros prévios de ataques obtiveram uma precisão superior a 80%. A medida de incerteza, calculada pela divergência Jensen-Shannon, manteve-se estável e controlada com o uso do K-greedy.

Este estudo apresenta uma estrutura sólida para a detecção de botnets em dispositivos IoT, oferecendo uma base valiosa com datasets e frameworks que podem ser úteis para a implementação do nosso projeto. O uso do algoritmo de agregação K-greedy traz inovações interessantes ao lidar com a incerteza dos dados, especialmente em ataques zero-day. No entanto, em nosso trabalho, não pretendemos seguir essa abordagem, já que a complexidade do K-greedy e seu foco em aprendizado bayesiano profundo não se alinham diretamente com nossas necessidades.

3.3 Securing Heterogeneous IoT With Intelligent DDoS Attack Behavior Learning

A proposta deste artigo é apresentar o FOGshield de Dao *et al* [2], um framework para prevenção de ataques DDoS em sistemas IoT. Ele utiliza fog computing e redes neurais auto-organizáveis (SOM) para detectar ameaças de forma eficiente. O FOGshield combina o processamento local dos dispositivos de nevoeiro com a aprendizagem federada para mitigar ataques de forma colaborativa. Essa abordagem permite uma detecção rápida e distribuída, aumentando a resiliência do sistema contra ataques DDoS.

O FOGshield consiste em um orquestrador central e múltiplos defensores de endpoint localizados na borda de cada sistema IoT homogêneo. O algoritmo SOM, uma técnica de aprendizado não supervisionado, é utilizado para transformar um espaço de entrada de alta dimensão em uma representação de menor dimensão, preservando a topologia dos dados. No contexto do FOGshield, o SOM classifica o tráfego de rede como normal ou DDoS. Os endpoints treinam seus mapas SOM localmente, usando dados específicos de suas redes, enquanto o orquestrador centraliza e aprimora os modelos SOM para uma colaboração mais eficaz na detecção de ataques.

O desempenho do FOGshield foi comparado com os frameworks MLDMF (Multilevel DDoS Mitigation Framework) e D-SOM (DDoS Prevention System). O FOGshield apresentou um desempenho superior aos frameworks comparados, com alta precisão (99,4%) e taxa de detecção (99,3%), devido à separação no treinamento dos filtros SOM para tráfego local. A arquitetura distribuída proporcionou menor latência e consumo de recursos (10% menos que abordagens distribuídas), enquanto o uso de CPU foi mantido em níveis

aceitáveis (35%). Em contraste, o MLDMF apresentou um alto uso de CPU (83%) e maiores gargalos devido ao processamento centralizado.

A proposta do FOGshield, com seu uso de fog computing para detectar e mitigar ataques DDoS, é particularmente interessante pela maneira como distribui o processamento, algo que pode ser adaptado para a nossa implementação. Embora o algoritmo SOM utilizado no FOGshield não esteja nos nossos planos, o framework oferece uma arquitetura flexível e eficiente, que pode ser aplicada em nosso trabalho. A estrutura descentralizada do FOGshield, especialmente no que diz respeito ao uso do aprendizado federado se alinham com os objetivos do nosso projeto. Portanto, o FOGshield apresenta um ponto de partida a implementação do nosso sistema, mesmo que ajustes sejam necessários para adaptar a técnica ao nosso foco.

3.4 Evading botnet detectors based on flows and Random Forest with adversarial samples

Neste artigo de Apruzzese e Colajanni [15], é apresentada uma análise experimental sobre a fragilidade de sistemas de detecção de intrusões na rede baseados em algoritmos de aprendizado de máquina diante de ataques adversários. O foco é a vulnerabilidade de um detector de botnet que utiliza o classificador random forest para inspecionar fluxos de rede. A pesquisa destaca como esses sistemas, apesar de eficientes, podem ser facilmente comprometidos por pequenas alterações nos padrões de comunicação maliciosos. O estudo busca abrir caminho para futuras soluções que fortaleçam esses sistemas, tornando-os mais robustos contra ataques adversários.

O trabalho considera dois modelos principais: o defensivo e o de ataque. O detector de botnet, baseado em random forest, é treinado para diferenciar entre fluxos legítimos e maliciosos, identificando variantes específicas de botnets. A estratégia do atacante envolve pequenas modificações nas sequências de comunicação entre os bots e a infraestrutura de comando e controle. Essas alterações, como leves aumentos na duração dos fluxos, bytes trocados e pacotes transmitidos, induzem o detector a classificar erroneamente fluxos maliciosos como legítimos.

Os experimentos demonstraram que pequenas modificações nas características dos fluxos de rede, como o aumento da duração da comunicação ou a adição de bytes e pacotes, podem reduzir drasticamente a taxa de detecção do detector de botnet. Em alguns casos, a detecção caiu de mais de 99% para menos de 1%. Esses resultados evidenciam a vulnerabilidade dos classificadores de random forest a ataques adversários, revelando a

necessidade de melhorias que tornem os sistemas de detecção de intrusões mais resistentes a essas manipulações.

Embora nossa implementação atual não utilize o classificador random forest, a análise deste estudo oferece uma perspectiva importante que pode ser aplicada em trabalhos futuros. As vulnerabilidades exploradas, como a manipulação de características dos fluxos de rede para burlar detectores, podem ser relevantes para outras técnicas de aprendizado de máquina que venhamos a adotar. Para este projeto, não pretendemos usar random forest, mas considerações sobre as adaptações dos ataques adversários devem ser investigadas em trabalhos futuros. Incorporar essas lições pode nos ajudar a criar soluções mais robustas e menos suscetíveis a pequenas manipulações de padrões de rede.

3.5 A new method for flow-based network intrusion detection using the inverse Potts model

Neste trabalho de Pontes, Marotta,*et. al* [9], apresentamos o Classificador de Fluxo Baseado em Energia (EFC), inspirado no modelo de Potts inverso da mecânica quântica e adaptado para a classificação de fluxos de rede. O EFC realiza a detecção de anomalias utilizando apenas fluxos benignos para treinar o modelo, permitindo identificar fluxos maliciosos com base em diferenças em relação aos padrões normais. A energia de cada fluxo é calculada com base em acoplamentos e campos locais entre suas características. Fluxos que seguem padrões benignos apresentam energias baixas, enquanto aqueles que se desviam desses padrões possuem energias mais altas, possibilitando uma distinção eficiente entre fluxos benignos e maliciosos.

O funcionamento do EFC envolve calcular a energia de um fluxo utilizando um modelo estatístico derivado de fluxos benignos. Conceitos do modelo de Potts inverso são aplicados para mapear as características dos fluxos em um grafo, determinando a energia associada a cada fluxo com base nos acoplamentos e campos locais. A classificação é realizada ao comparar a energia calculada com um limite pré-definido, identificando fluxos que não correspondem aos padrões benignos como potencialmente maliciosos. Esse método é eficaz para detectar anomalias mesmo na ausência de amostras de ataques conhecidos.

Os resultados mostram que o EFC alcança desempenho notável na classificação de tráfego de rede, com valores de F1 score de até 97% e AUC de até 99%, que são comparáveis aos obtidos por algoritmos tradicionais de aprendizado de máquina, como k-vizinhos mais próximos e árvores de decisão. Além disso, o EFC demonstra uma superior capacidade de adaptação a novos domínios, superando outros classificadores em termos de F1 score

e se mantendo entre os melhores em AUC. Estes resultados indicam que o EFC não só é eficaz na detecção de fluxos maliciosos, mas também oferece vantagens significativas em cenários variados.

O uso do Energy-based Flow Classifier (EFC) em nosso trabalho oferece uma solução eficiente para a detecção de botnets, adaptando-se a diferentes domínios sem a necessidade de conhecer o comportamento de fluxos maliciosos previamente. Ao integrar o aprendizado federado, podemos realizar a detecção de forma descentralizada, garantindo privacidade e reduzindo o uso de recursos. Com um desempenho competitivo em comparação com classificadores tradicionais, o EFC se destaca pela simplicidade e eficácia, sendo ideal para ambientes com recursos limitados. Essa abordagem proporciona uma solução escalável e robusta para detecção de botnets, alinhada aos nossos objetivos de implementação.

3.6 Botnet detection based on network flow analysis using inverse statistics

O trabalho de Lopes, Daniele A. G. *et al.* [10] propõe um método para identificar *botnets* por meio da análise de fluxos de rede. O enfoque principal é a aplicação do classificador EFC (Energy Flow based Classifier), utilizando estatísticas inversas que destacam comportamentos anômalos em grandes volumes de dados de tráfego de rede. A técnica se baseia na observação de padrões de comunicação entre os *bots* e seus servidores de comando e controle. Ao analisar fluxos de rede e suas variações, o modelo consegue distinguir tráfego malicioso de tráfego legítimo.Utilizando a análise de fluxos de rede combinada com estatísticas inversas para detectar *botnets*.

Primeiramente, os fluxos de rede são coletados e processados, sendo representados por atributos como endereços IP, portas, protocolos e volume de tráfego. A partir desses dados, são extraídas distribuições de variáveis relacionadas ao comportamento da comunicação de rede. Em seguida, as estatísticas inversas são aplicadas para identificar desvios de comportamento em relação ao tráfego esperado, destacando assim as anomalias. O método busca identificar padrões característicos de *botnets*, como comunicações regulares e sincronizadas entre dispositivos infectados e seus servidores de controle. A técnica é projetada para detectar *botnets* de diferentes tamanhos, incluindo aquelas em fases iniciais de operação. Após a identificação de possíveis *botnets*, o modelo passa por uma validação com dados reais e simulados para garantir sua eficácia.

Os resultados do trabalho mostram a eficácia do classificador EFC na detecção de tráfego relacionado a atividades de botnets. O EFC utiliza amostras de fluxos benignos para

inferir um modelo estatístico durante o treinamento e, em seguida, calcula as energias de amostras benignas e maliciosas para classificá-las. Nos testes intra-domínio com os conjuntos de dados ISOT HTTP e CTU-13, o EFC apresentou boa separação entre os fluxos benignos e maliciosos, com um F1-score acima de 0.98 e AUC acima de 0.99. No teste inter-domínio, em que o treinamento foi realizado com o ISOT HTTP e o teste com o CTU-13, o EFC obteve melhor desempenho em comparação aos outros classificadores, alcançando um F1-score de 0.663. Classificadores como NB, RF e AD tiveram F1-scores baixos devido à classificação incorreta de instâncias maliciosas como benignas.

Este trabalho oferece uma base sólida para a pesquisa de detecção de *botnets* utilizando o *Energy-based Flow Classifier (EFC)*, fornecendo um modelo adaptável a diferentes domínios e com boa capacidade de detectar *botnets* em cenários complexos. Ao integrar o EFC com treinamento federado, a pesquisa poderia avançar significativamente, criando modelos mais resilientes, com melhor generalização para diferentes redes e contextos. O treinamento federado é vantajoso em ambientes onde os dados são distribuídos de forma heterogênea. O EFC, já demonstrando baixa sensibilidade a variações na distribuição de dados, pode se beneficiar dessa abordagem colaborativa, consolidando diferentes distribuições de tráfego malicioso.

3.7 Botnet Intrusion Detection Method based on Federated Reinforcement Learning

A proposta do trabalho de Lou, Xingyu *et al.* [7] é desenvolver um método eficiente de detecção de intrusões de botnets utilizando uma combinação de aprendizado federado e aprendizado por reforço. O aprendizado federado permite que múltiplos dispositivos participem do treinamento de um modelo global sem a necessidade de compartilhar diretamente seus dados, o que é crucial para lidar com a heterogeneidade dos dados distribuídos em diferentes redes. No entanto, o desafio de comunicação frequente entre os nós e o servidor central, causado pela distribuição desigual dos dados, é abordado com a introdução de uma estratégia de seleção inteligente de nós baseada em aprendizado por reforço. Essa estratégia identifica os nós mais adequados para participar do treinamento global, reduzindo os custos de comunicação e melhorando a estabilidade do treinamento. Como resultado, a proposta visa aumentar a precisão e a eficiência da detecção de botnets em redes complexas e heterogêneas, resolvendo os problemas típicos dos métodos tradicionais de detecção.

Em vez de centralizar todo o treinamento, o aprendizado federado distribui o processo entre diversos nós locais que treinam o modelo usando seus próprios dados e enviam os

parâmetros ao servidor central, que os agrega. Em vez de selecionar clientes de forma aleatória (o que pode prejudicar a performance do modelo global), o servidor central utiliza um algoritmo de aprendizado por reforço para calcular o valor de cada cliente com base nos pesos do modelo. O DQN, uma versão aprimorada do algoritmo Q-Learning, usa uma rede neural para calcular o valor de cada ação. Após cada rodada de treinamento, ele ajusta os pesos do modelo com base no desempenho e recompensa obtidos, maximizando as recompensas cumulativas ao longo do treinamento. Essa abordagem reduz o tempo de treinamento e os custos de comunicação, além de aumentar a precisão e a robustez do modelo para detectar botnets.

Os resultados mostraram que, embora o treinamento com dados Non-IID apresentasse flutuações e dificuldades de convergência, os modelos de aprendizado federado, especialmente o GRU otimizado com Reforço Profundo, superaram os modelos LSTM em termos de precisão e estabilidade. O modelo GRU com aprendizado federado e otimização por reforço (FRL-GRU) alcançou uma convergência mais rápida e precisa, exibindo uma curva de precisão ascendente estável e uma perda global menor. Por outro lado, o modelo LSTM apresentou maiores flutuações e um tempo de treinamento mais longo. A análise também mostrou que a otimização por reforço melhorou significativamente a estabilidade e a velocidade de treinamento, particularmente com dados Non-IID, onde a seleção inteligente de nós mitigou os efeitos da distribuição desigual dos dados. O modelo FRL-GRU demonstrou a melhor combinação de precisão, estabilidade e eficiência no treinamento.

Esse trabalho fornece avanços importantes ao introduzir a técnica de aprendizagem por reforço federado para melhorar o processo de treinamento em cenários Non-IID. Além de superar o problema das variações de dados entre diferentes nós usando aprendizagem por reforço para decidir quais deles irão participar no treinamento global. Dessa forma, a precisão e a estabilidade do modelo são melhoradas. O FRL reduz o custo de comunicação e acelera a convergência do modelo em comparação com as técnicas convencionais de aprendizado federado, demonstrando que o aprendizado profundo pode resultar em modelos mais resilientes e estáveis.

3.8 Hybrid Deep Learning for Botnet Attack Detection in the Internet-of-Things Networks

O trabalho de Popoola, Segun I. *et al.* [11] propõe métodos avançados de aprendizado profundo para melhorar a detecção de ataques botnet em redes IoT. Devido às limitações de memória e processamento dos dispositivos IoT, métodos tradicionais de DL, que requerem grandes volumes de dados e recursos computacionais elevados, não são adequados

para esses ambientes. Visando superar as limitações dos métodos tradicionais, reduzindo a complexidade computacional e melhorando a precisão na detecção de botnets em redes IoT. Para enfrentar o desafio associados ao processamento de dados de alta dimensionalidade é sugerido o uso de duas abordagens LAE, e BLTM.

Autoencoder de Memória de Longo Prazo e Curto Prazo (LAE) é o método é utilizado para reduzir a dimensionalidade dos dados de rede ao criar uma representação de espaço latente de baixa dimensão, o que facilita o processamento e a análise de dados complexos em dispositivos IoT com recursos limitados. Long Short-Term Memory Bidirecional (BLSTM) é uma técnica de aprendizado profundo é proposta para aprender representações hierárquicas das características dos dados e mudanças inter-relacionadas a partir dos dados brutos, oferecendo uma maneira eficaz de classificar a representação de espaço latente dos dados de tráfego de rede.

O estudo avaliou a eficácia de vários algoritmos de otimização em dois contextos principais: redução da dimensionalidade das características e classificação de tráfego de rede usando modelos de aprendizado profundo. Em termos de redução da dimensionalidade, o algoritmo Adam demonstrou o melhor desempenho ao treinar o modelo LAE, apresentando a menor perda de reconstrução, variando entre 0.0041 e 0.0023.No que diz respeito à classificação de tráfego de rede em cenários binários, o modelo BLSTM profundo apresentou as menores perdas de treinamento e validação quando treinado com o otimizador Nadam. No que diz respeito à classificação de tráfego de rede em cenários binários, o modelo BLSTM profundo apresentou as menores perdas de treinamento e validação quando treinado com o otimizador Nadam.

A aplicação de métodos de redução de dimensionalidade e otimização ajuda a melhorar a eficiência dos modelos federados, permitindo que eles sejam treinados em dispositivos com limitações de memória e processamento, enquanto ainda conseguem detectar ataques com precisão. O estudo demonstra como o Autoencoder de Longo e Curto Prazo (LAE) pode reduzir eficazmente a dimensionalidade dos dados, facilitando o uso de técnicas de aprendizado profundo em dispositivos IoT com recursos limitados. Assim a aplicação do autoencoder LSTM (LAE) e do modelo BLSTM para redução da dimensionalidade e classificação fornece uma solução que otimiza o desempenho dos modelos federados, garantindo uma melhor utilização dos recursos limitados dos dispositivos IoT.

3.9 FLEAM: A Federated Learning Empowered Architecture to Mitigate DDoS in Industrial IoT

O trabalho de Li, Jianhua *et al.* [6] propõe uma arquitetura chamada FLEAM, que combina computação em névoa (*Fog*) e aprendizagem federada (FL) para mitigar ataques de DDoS em sistemas da Internet das Coisas Industrial (IIoT). A principal vantagem dessa abordagem é a distribuição da inteligência de mitigação ao longo das rotas de ataque, o que aumenta o custo para os atacantes e permite detecções mais rápidas e precisas. Ao contrário dos modelos tradicionais, que centralizam a mitigação ao redor da vítima, FLEAM promove uma colaboração entre defensores, visando aumentar a eficácia e reduzir a latência na resposta a ataques. A arquitetura utiliza o protocolo IMA-GRU para processar dados distribuídos e ajustar modelos de maneira dinâmica, enfrentando a rápida evolução das ameaças no ambiente IIoT.

Além disso, o FLEAM foi projetado para superar as limitações das soluções atuais, como a falta de colaboração entre defensores. A computação em névoa atua distribuindo recursos computacionais ao longo da rede, mais próximos das fontes de dados, o que reduz a latência e permite uma resposta mais rápida aos ataques. A névoa é responsável por processar dados localmente, fornecendo maior proximidade às ameaças, o que facilita a detecção e resposta em tempo real. A aprendizagem federada complementa essa estrutura ao permitir que os dispositivos colaborem na criação de um modelo global de detecção de ataques sem precisar compartilhar os dados brutos entre eles. Cada nó na rede, que inclui dispositivos de névoa, treina localmente com seus próprios dados e envia apenas os parâmetros de atualização do modelo para um servidor central. Assim, é possível criar um modelo atualizado e mais preciso, que reflete as condições dinâmicas do ambiente IIoT

O estudo sobre o FLEAM demonstrou que a arquitetura supera as soluções clássicas na mitigação de ataques DDoS, com menor atraso de download e upload, além de melhorar o uso da largura de banda e a taxa de transferência de pacotes. O atraso de download, por exemplo, foi reduzido para entre 0,7 e 5 segundos com FLEAM, comparado a 6 a 10 segundos em soluções clássicas. Além disso, a fila de espera no tráfego benigno foi minimizada, e o consumo de largura de banda foi otimizado ao bloquear códigos DDoS antes de atingirem a vítima. Em termos de precisão, FLEAM alcançou 98% de acurácia em mitigações conjuntas, superando soluções individuais que tiveram cerca de 51% de acurácia. Esses resultados mostram que o FLEAM é mais eficaz, rápido e preciso, forçando atacantes a gastarem mais recursos, o que desestimula novos ataques.

No FLEAM, FL é permitido que dispositivos colaborem no treinamento de modelos sem compartilhar dados brutos, o que aumenta a privacidade e reduz a latência, algo essencial

para a detecção de anomalias em sistemas de energia. Além disso, a computação em névoa no FLEAM processa informações localmente, o que também pode acelerar a resposta em classificadores EFC ao identificar padrões anômalos em fluxos de energia. Ao lidar com dados heterogêneos e dinâmicos, o FLEAM oferece robustez, o que é crucial para sistemas de energia em constante mudança. Por fim, a colaboração multi-defensores do FLEAM pode ser aplicada ao classificador EFC, onde diferentes dispositivos compartilham informações sobre anomalias energéticas, melhorando a precisão da mitigação e tornando os sistemas mais resilientes.

3.10 In-Depth Feature Selection for the Statistical Machine Learning-Based Botnet Detection in IoT Networks

O artigo [16], por Kalakoti, Rajesh, Nõmm, Sven e Bahsi, Hayretdin, aborda o uso da Internet das Coisas (IoT) no cotidiano e os desafios de segurança associados ao crescimento exponencial desses dispositivos, como ataques de botnets. O ciclo de vida das botnets possui quatro fases: formação, comando e controle (C&C), ataque, e pós-ataque, cada uma com diferentes características e vulnerabilidades. Uma das principais soluções de segurança sugeridas é a detecção comportamental e anomalias em vez de assinaturas tradicionais, permitindo a identificação de padrões maliciosos sem a necessidade de assinaturas conhecidas.O estudo foca em melhorar a detecção de tráfego malicioso em redes IoT por meio de técnicas de seleção de características. O objetivo é encontrar um subconjunto ideal de características que otimize a performance de diferentes classificadores de machine learning, como random forest e k-nearest neighbor, usando métodos de filtro e wrapper.

Os modelos de filtro avaliam características com base em critérios discriminatórios sensíveis à classe, sem depender do algoritmo de classificação. Dentro desse grupo, destacam-se técnicas como a correlação de Pearson, o *Fisher score*, a informação mútua e a análise de variância. Os métodos de *wrapper*, por outro lado, são específicos para o algoritmo de classificação utilizado. No estudo, foram testadas seis técnicas de *wrapper*, com destaque para a Eliminação Recursiva de Recursos (RFE), Seleção Sequencial para Frente (SFS) e Seleção Sequencial para Trás (SBS), que mostraram melhores resultados.

Os resultados do estudo mostraram que a detecção de ataques de *botnet* em IoT, utilizando os conjuntos de dados N-BaIoT e MedBIoT, alcançou mais de 99% de precisão com apenas 3 a 7 características selecionadas. A combinação do método de seleção de caracte-

rísticas Seleção Sequencial para Trás (SBS) com o classificador de árvore de decisão (DT) apresentou o melhor equilíbrio entre capacidade de detecção e custo computacional. Os modelos baseados em árvores (DT, *Extremely Randomized Trees* e *Random Forest*) superaram os demais, especialmente em classificações multiclasses, enquanto o classificador k-NN foi menos eficaz e exigiu mais tempo computacional. Os resultados demonstraram que a metodologia proposta superou modelos existentes em termos de precisão e pontuação F1, ressaltando a eficácia da seleção estratégica de características para melhorar a detecção em sistemas de intrusão em ambientes IoT.

Ao demonstrar a eficácia da seleção de características na melhoria do desempenho de modelos de classificação. A EFC, que se concentra no fluxo de energia em sistemas IoT, pode se beneficiar de métodos de seleção de características que identificam as variáveis mais relevantes, otimizando o uso de recursos limitados em dispositivos com capacidade de processamento e energia restritas. Além disso, a comparação entre métodos de seleção (filtro e wrapper) e a análise das diferentes características destacam a importância de entender o contexto específico dos dados de energia em um ambiente federado. Isso pode levar à construção de modelos mais eficientes e adaptáveis, permitindo que os dispositivos federados processem informações de maneira mais eficaz, minimizando o consumo de energia enquanto mantêm altos níveis de precisão na detecção de padrões ou anomalias.

Neste Trabalho será utilizado a técnica de treinamento federado em conjunto com o classificador baseado no fluxo de energia (*EFC*) para realizar a detecção de ataques de *botnets*. No ambiente de aprendizado federado o EFC torna mais eficiente e adaptável em sua detecção, permitindo um consumo menor de energia com seu alto grau de detecção de anomalias. No projeto é utilizado os conjuntos de dados CTU-13 e o ISOT HTTP para o experimento. Por fim serão comparados os resultados deste trabalho com os resultados de [9], onde é usado o EFC sem o treinamento federado.

3.11 Network Intrusion Detection Scheme based on Federated Learning in Heterogeneous Network Environments

No trabalho de Yuedi Zhu, Chao Li e Yong Wang [4], é proposto uma hibrida solução usando aprendizado federado com o Modelo Gaussian Mixture (GMM) para a detecção de intrusão de rede em dispositivos de bordas. Nele aborda o desafio da generalização que o aprendizado federado enfrenta sobre a capacidade de processamento limitada dos disposi-

tivos, e a ameaça potencial de ataques de envenenamento. Mostrando a preocupação em alcançar um modelo generalizado em ambientes de rede reais e complexos. Para mitigar esse problema é usado o modelo FedAdagrad com o GMM, integrando com o Energy Flow Classifier (EFC) na arquitetura para dados heterogêneos.

Em sua metodologia, antes de iniciar o processo de aprendizado federado, o EFC foi treinado com o conjunto de dados de treino para obter os valores de energia dos conjuntos de treino e teste, os quais foram adicionados como uma característica ao dataset original. Em seguida, 10% da partição de treino foi reservada para ajuste e validação do modelo. Posteriormente, o GMM foi utilizado para o treinamento local, permitindo calcular a densidade de probabilidade de cada amostra pertencer à distribuição de dados normais, onde é identificada como anômala se sua densidade estiver abaixo do limiar. Durante a inicialização do processo de aprendizado federado, cada cliente obtém a arquitetura do modelo do servidor global e inicializa seus pesos aleatoriamente, iniciando então o FedAdagrad. Cada participante local atualiza o modelo utilizando a otimização Adagrad e envia seus parâmetros atualizados ao servidor central. Após receber os parâmetros e a soma do histórico de gradientes de cada cliente, o servidor central realiza a agregação por meio da média ponderada. A cada época, os clientes treinam seus dados locais com a versão mais recente do modelo global.

Nos experimentos, foram utilizadas diferentes estratégias de aprendizado federado com o GMM, comparando o FedAdagrad com o FedAvg e o FedAdma. Os resultados mostram que, sem a utilização do EFC, há uma flutuação na métrica F1-score ao longo de dez rodadas. Verifica-se que a estratégia de aprendizado federado com EFC apresenta desempenho significativamente superior em ambientes heterogêneos, destacando sua importância para a generalização em redes com essas características. Outro teste realizado comparou o Isolation Forest (IF) e o Local Outlier Factor (LOF), utilizando como métricas o F1-score Local e o F1-score Cruzado. A análise demonstrou um desempenho superior em ambas as métricas, além de apresentar um desvio padrão menor entre os clientes, indicando um desempenho mais consistente e estável.

Este estudo oferece uma base sólida sobre como o EFC pode ser utilizado em metodologias de aprendizado federado, demonstrando um Sistema de Detecção de Intrusão em Redes (NIDS) eficiente e robusto em redes heterogêneas. Apresenta métricas relevantes para o nosso trabalho, servindo como base de comparação. Além disso, demonstra uma alternativa de uso do EFC em ambientes de aprendizado federado, em relação à abordagem adotada em nosso estudo.

3.12 Discussão

A integração do classificador baseado no fluxo de energia (EFC) com o paradigma de aprendizagem federada se alinha a diversas propostas recentes da literatura que buscam formas mais seguras, escaláveis e eficientes para detectar ataques em redes IoT. Trabalhos como os de Popoola et al. [11] e Zhu et al. [4] demonstram que o uso do aprendizado federado permite treinar modelos robustos mesmo em cenários onde o compartilhamento de dados é inviável ou indesejado. Esses estudos reforçam a ideia de que a descentralização do aprendizado pode preservar a privacidade sem sacrificar desempenho, especialmente quando associada a mecanismos capazes de lidar com a heterogeneidade dos dados, como o FedAdagrad ou a seleção de características relevantes. A escolha do EFC neste trabalho, portanto, aproveita essa perspectiva, oferecendo uma alternativa leve e compatível com dispositivos de borda, e ao mesmo tempo robusta contra variações nos dados.

A proposta deste trabalho também dialoga com iniciativas que visam melhorar a agregação entre modelos locais, como o K-greedy (Zhang et al., [3]) e o FedAvg (McMahan et al., [8]), ambos amplamente utilizados em contextos de redes IoT heterogêneas. Enquanto o K-greedy procura selecionar modelos com maior incerteza para otimizar a diversidade informacional, o FedAvg aposta na média ponderada das atualizações locais como uma forma simples e eficaz de construir um modelo global. No experimento descrito anteriormente, a variação de pesos na combinação de modelos EFC mostrou que, mesmo com dispositivos de baixa acurácia, é possível alcançar ganhos quando há uma calibragem cuidadosa das contribuições individuais. Isso sugere que, embora técnicas mais sofisticadas como o K-greedy tragam benefícios teóricos, abordagens ponderadas bem ajustadas também oferecem excelente desempenho prático, especialmente quando se leva em conta a simplicidade computacional e a facilidade de implementação.

Outro ponto de contato importante está nos estudos que lidam com ataques adversários ou técnicas de mitigação distribuída, como nos trabalhos de Apruzzese & Colajanni [15], Dao et al. [2] e Li et al. [6]. Esses estudos apontam para a necessidade de resiliência diante de adversários inteligentes e a importância de arquiteturas que respondam com agilidade a eventos maliciosos. A arquitetura federada do presente experimento, somada à natureza estatística do EFC, é promissora nesse sentido, pois evita a dependência de assinaturas fixas e permite respostas descentralizadas e adaptativas. Ainda que o classificador EFC, por si só, não seja explicitamente projetado para resistir a ataques adversários, seu uso em contexto federado, com pesos distribuídos de maneira controlada, pode atenuar os impactos de fontes comprometidas, além de se beneficiar de futuras estratégias como validação cruzada entre dispositivos confiáveis.

Por fim, os trabalhos de Kalakoti et al. [16] e Pontes et al. [9] reforçam a relevância do uso do EFC em contextos de detecção de anomalias, destacando sua capacidade de identificar fluxos maliciosos mesmo sem contato prévio com padrões de ataque. Os experimentos inter-domínio mostraram que o EFC mantém desempenho elevado, especialmente em F1-score e AUC, mesmo diante de variações nos dados. No contexto do aprendizado federado, essas características são ainda mais valiosas, pois garantem que dispositivos com contextos locais diferentes ainda possam contribuir para um modelo global robusto. A combinação entre o EFC e o aprendizado federado, portanto, se mostra coerente com o estado da arte e indica um caminho sólido para detectar botnets em redes IoT com baixo custo computacional, alta adaptabilidade e respeito à privacidade dos dados.

Capítulo 4

Arquitetura de aprendizado federado utilizando EFC

Neste capítulo, é apresentada a proposta de uma arquitetura baseada em aprendizado federado voltada à detecção de botnets em redes. O objetivo é construir um sistema colaborativo, no qual múltiplos dispositivos possam treinar localmente seus modelos sem a necessidade de compartilhar dados, por meio da integração do classificador Energy-based Flow Classifier (EFC) em um cenário federado, conforme ilustrado na Figura 4.1. A estrutura é composta por três blocos principais: os dispositivos locais, o EFC global, com a comunicação realizada por meio da troca das matrizes de acoplamento e dos campos locais, e o bloco do *Dataset*. Os dispositivos locais e o EFC global também apresentam blocos internos, cujo funcionamento será detalhado nas seções seguintes.

Cada dispositivo participante da arquitetura possui o seu próprio conjunto de dados local (bloco *Dataset*). O conjunto é utilizado tanto para o treinamento quanto para a validação dos resultados. Essa separação de dados por cliente reforça o conceito de aprendizado federado, onde nenhuma amostra de dados é compartilhada entre os dispositivos ou com o servidor global. A variação natural entre as bases de dados de cada dispositivo permite ter cenários com características próprias, contendo distribuição de classes e volume de dados desbalanceados entre eles.

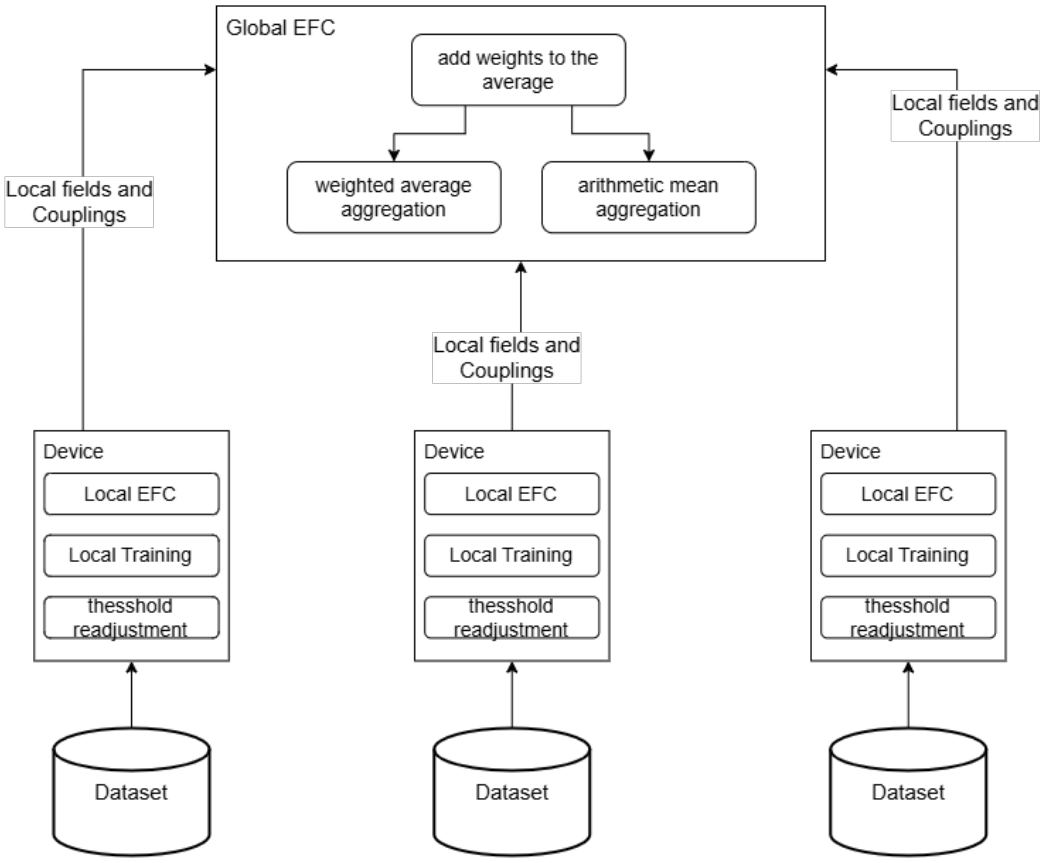

Figura 4.1: Arquitetura de aprendizado federado utilizando EFC

Em cada cliente (*Device*) executa três etapas principais. A primeira é o bloco que representa o *LocalEFC*, nele cada cliente mantém uma instância local do classificador EFC, com seus próprios hiperparâmetros. Essa inicialização permite que cada dispositivo aprenda de forma independente sobre os seus próprios dados. Em seguida cada cliente realiza o treinamento do seu modelo EFC utilizando exclusivamente o seu conjunto de dados local, o EFC nesta pate do treinamento só utiliza dados benignos dele.

Na arquitetura proposta, os únicos elementos enviados de cada cliente ao servidor global são a matriz de acoplamento e os campos locais, ambos aprendidos localmente durante o treinamento do classificador EFC. Essa escolha está diretamente relacionada ao funcionamento interno do modelo Energy-based Flow Classifier (EFC), cujo processo de inferência é fundamentado no cálculo da energia associada a cada fluxo de rede, demostrado em 2.7. Dessa forma, ao agregá-los em nível global, é possível reconstruir uma representação coletiva do conhecimento aprendido por todos os clientes. Assim, a seleção dos dois parâmetros como elementos de agregação não só preserva a natureza distribuída do

aprendizado, como também mantém a coerência matemática do modelo, viabilizando a inferência global baseada em energia mesmo em um ambiente federado.

O *GlobalEFC* é responsável por realizar a agregação dos parâmetros locais recebidos com o objetivo de gerar um modelo mais robusto e representativo. Em vez de centralizar os dados brutos, o *GlobalEFC* recebe de cada cliente apenas os parâmetros estruturais do classificador: a matriz de acoplamento e os campos locais. Para combinar esses parâmetros, a arquitetura propõe dois métodos de agregação: média ponderada e média aritmética. Ambas serão avaliadas separadamente, com o objetivo de comparar seus desempenhos.

4.1 Média Ponderada

Nesse método, cada conjunto de parâmetros locais é multiplicado por um peso antes da soma, e o valor final é normalizado pela soma total dos pesos. Os pesos são definidos manualmente e representam a influência relativa de cada cliente na formação do modelo global. A introdução dos pesos permite avaliar o impacto da contribuição diferenciada de clientes com desempenho variável. Por exemplo, se um cliente alcançou uma alta taxa de acerto em sua avaliação local, ele pode receber um peso maior, enquanto outro com desempenho inferior pode receber um peso menor. Essa configuração permite analisar se dar mais influência a um cliente com melhor desempenho local melhora a performance global. O cálculo da média ponderada segue a fórmula:

$$\text{Parâmetro Global} = \frac{\sum_{i=1}^{N} w_i \cdot \theta_i}{\sum_{i=1}^{N} w_i} \tag{4.1}$$

Onde w_i é o peso atribuído ao cliente i, o θ_i representa os parâmetros locais (matriz de acoplamento e campos locais), e o N é o número total de clientes. Nessa situação torna o impacto de cada cliente proporcional ao seu peso em relação aos demais.

4.2 Média Aritmética

Nesta abordagem, os parâmetros de cada cliente (matriz de acoplamento e campos locais) são multiplicados por um peso pré-definido. Representando a influência relativa daquele cliente. Em seguida, esses valores ponderados são somados e divididos pelo número total de dispositivos. Mostrado na equação 4.2,

$$\text{Parâmetro Global} = \frac{1}{N} \sum_{i=1}^{N} w_i \cdot \theta_i \tag{4.2}$$

essa estratégia permite explorar o impacto dos pesos, mas mantém uma normalização fixa, tratando todos os dispositivos igualmente no denominador. Esse mecanismo de pesos nas duas agregações com a média, fornece flexibilidade à arquitetura, permitindo realizar experimentos com diferentes configurações de influência entre os clientes (ex.: 9-1, 8-2, 7-3...), com o objetivo de investigar quais estratégias de ponderação levam a melhores resultados de classificação no modelo global.

Após a agregação, os novos parâmetros globais são redistribuídos a todos os clientes, onde cada um deles então atualiza seus modelos locais com os parâmetros agregados recebidos, realiza o reajuste do limiar, recalculando seu critério de decisão com base nos novos parâmetros, posteriormente, realiza o teste local para avaliar melhorias no desempenho de classificação. O intuito de atualizar o limiar de decisão, vem do fato de que após a etapa de agregação no servidor global, em que as matrizes de acoplamento e os campos locais de todos os clientes são combinados para formar um conjunto unificado de parâmetros, cada cliente recebe de volta os parâmetros globais do modelo EFC. No entanto, como o limiar de decisão (*threshold*) utilizado pelo EFC está diretamente relacionado à distribuição das energias das amostras de treinamento, é necessário realizar um reajuste local desse threshold.

Esse ajuste é essencial porque o limiar original foi calculado com base na distribuição de energia de cada modelo local, o que deixa de ser válido após a atualização dos parâmetros com os valores globais. O cálculo do novo limiar segue o mesmo princípio da função 2.7. Nesse processo consiste em reaplicar o modelo EFC, agora com os parâmetros globais sobre o conjunto de dados de treinamento local de cada cliente. Calculando novamente a energia de cada instância utilizando os novos parâmetros, ordenar os valores de energia e selecionar aquele que corresponde ao quantil definido, e atualizar o limiar local com esse novo valor, tornando o modelo local compatível com o novo estado global do classificador.

Capítulo 5

Metodologia

Nesta seção, são descritos os procedimentos metodológicos adotados para a implementação e avaliação da proposta de detecção de botnets por meio do aprendizado federado utilizando o classificador Energy-based Flow Classifier (EFC). O objetivo é descrever o funcionamento do sistema, desde o particionamento dos dados, passando pelo treinamento local em dispositivos distribuídos, até o processo de agregação dos parâmetros no servidor central e o reajuste local do limiar de decisão (*threshold*). A seção será dividida em dois cenários, cada um representando a quantidade de dispositivos utilizados nos testes. Para cada cenário, foram utilizados os conjuntos de dados CTU-13 e ISOT-HTTP separadamente, com foco em observar os diferentes resultados com bases de dados distintas.

5.1 Cenários

5.1.1 Cenário dos dois dispositivos

No primeiro cenário, o experimento foi realizado com dois dispositivos (clientes), onde cada cliente mantém uma instância local do classificador EFC, com seus próprios hiperparâmetros. Optou-se por utilizar as configurações padrão do EFC, conforme apresentado na Tabela 5.1.

Tabela 5.1: Configuração Padrão do EFC

hiperparâmetros	**Valores**
pseudocounts	0.5
cutoff_quantile	0.95
n_bins	30
n_jobs	none

O parâmetro *pseudocounts* representa o peso das pseudocontagens adicionadas às frequências empíricas; *cutoff_quantile* refere-se ao quantil utilizado para definir o limiar de energia do modelo; *n_bins* indica o número de níveis ao discretizar os atributos dos dados; e, por fim, *n_jobs* corresponde ao número de execuções paralelas, sendo que o valor None equivale a 1.

Para cada dispositivo participante, o conjunto de dados foi dividido em porções distintas de treino e teste, conforme detalhado na Tabela 5.2.

Tabela 5.2: Divisão do Dataser para treino e teste cenário 1

Clientes	Treino	Teste
Dispositivo 1	25%	75%
Dispositivo 2	0.1%	99.9%

Essa divisão foi propositalmente desigual entre os dispositivos, de modo a simular cenários com diferentes capacidades de aprendizado local. Por exemplo, um dos dispositivos recebeu apenas 0.1% dos dados para treinamento e 99.9% para teste, enquanto outro recebeu 25% para treino e 75% para teste. O objetivo dessa configuração é criar um ambiente controlado onde seja possível avaliar se um dispositivo com desempenho local insatisfatório, devido à escassez de dados de treinamento, pode melhorar seu desempenho ao participar do processo de agregação federada, beneficiando-se da influência de um dispositivo com treinamento mais robusto. Dessa forma, a metodologia permite investigar a eficácia da aprendizagem colaborativa em compensar limitações locais.

Com a divisão dos dados concluída e o classificador EFC instanciado em cada dispositivo, inicia-se a etapa de treinamento local. Cada dispositivo treina seu modelo de forma independente, utilizando apenas os dados de treinamento disponíveis localmente. O EFC é configurado para realizar classificação binária, sendo a classe base definida como os fluxos benignos, por meio do parâmetro *base_class*. Essa configuração permite ao modelo estimar a energia de cada instância com base em sua compatibilidade com o padrão aprendido da classe considerada normal (fluxos benignos), permitindo assim identificar desvios que possam indicar comportamentos anômalos.

Após o treinamento local, e antes de qualquer processo de agregação federada, cada modelo é avaliado utilizando seu respectivo conjunto de teste. Essa avaliação preliminar, baseada em métricas como F1-score, tem como objetivo estabelecer uma linha de base de desempenho para cada cliente. Com isso, torna-se possível analisar, posteriormente, se houve melhoria no desempenho de dispositivos com treinamento limitado, após a influência dos parâmetros agregados no modelo global.

Finalizado o treinamento local, cada dispositivo envia ao servidor global os parâmetros aprendidos: a matriz de acoplamento e os campos locais, que são extraídos por meio da estrutura interna chamada *estimator*, presente no classificador EFC. Esses parâmetros representam o conhecimento aprendido localmente sobre os padrões de fluxo benigno. Então é realizada a etapa de agregação federada, onde os valores recebidos de cada cliente são combinados segundo dois métodos distintos: média aritmética e média ponderada. Em ambos os casos, os parâmetros de cada cliente são previamente multiplicados por um peso específico, definido conforme diferentes combinações preestabelecidas, na tabela 5.3:

Tabela 5.3: Combinações dos pesos

Dispositivo 1	Dispositivo 2
0.9	0.1
0.8	0.2
0.7	0.3
0.6	0.4
0.5	0.5
0.4	0.6
0.3	0.7
0.2	0.8
0.1	0.9

Essas combinações correspondem a diferentes níveis de influência relativa entre os clientes, variando desde um cenário em que um cliente domina quase completamente a agregação até uma situação de equilíbrio ou inversão. Para cada iteração de teste, é aplicada uma dessas configurações de pesos, permitindo a comparação do desempenho resultante em cada cenário. Após o cálculo da agregação (média aritmética ou ponderada), os novos parâmetros globais são definidos. Em seguida, esses parâmetros são enviados de volta aos dispositivos, onde seus modelos locais são atualizados com os valores recebidos.

Após a atualização dos modelos locais com os parâmetros agregados, torna-se necessário recalibrar o limiar de decisão (*threshold*) de cada dispositivo. Para isso, é utilizado o método interno *_define_cutoff()* do próprio EFC, acessado via *estimator*. Esse procedimento consiste em aplicar novamente o modelo, agora com os parâmetros globais recebidos, sobre o conjunto de dados de treinamento local. A partir disso, são recalculados os valores de energia das instâncias, que então são ordenados para identificar o valor correspondente ao quantil previamente definido (95%). Esse valor passa a ser adotado como o novo limiar de decisão, garantindo que o processo de decisão do modelo continue ajustado ao seu respectivo ambiente local, mesmo após a incorporação das informações globais.

Com todas as etapas concluídas, incluindo o treinamento local, a agregação dos parâmetros, a atualização dos modelos e o reajuste do limiar, é realizada a etapa final de avaliação. Na qual se compara o desempenho do dispositivo que apresentou resultado insatisfatório antes e depois da agregação. O objetivo é verificar se houve melhora significativa no desempenho desses dispositivos após a influência dos parâmetros agregados. Validando assim a eficácia do aprendizado federado com EFC como uma abordagem colaborativa para compensar limitações locais.

5.1.2 Cenário com quatro dispositivos

No segundo cenário, o experimento foi realizado com quatro dispositivos (clientes), sendo a metodologia aplicada semelhante à do primeiro cenário. Cada cliente foi instanciado localmente com o classificador EFC, utilizando os parâmetros padrão conforme a Tabela 5.1. Para cada dispositivo participante do aprendizado federado, o conjunto de dados foi dividido em partes distintas de treino e teste, conforme apresentado na Tabela 5.4.

Tabela 5.4: Divisão do Dataser para treino e teste no cenário 2

Clientes	Treino	Teste
Dispositivo 1	25%	75%
Dispositivo 2	0.1%	99.9%
Dispositivo 3	75%	25%
Dispositivo 4	0.1%	99.9%

Essa divisão segue o mesmo propósito do cenário 1, com o objetivo de apresentar diferentes casos com distintas capacidades de aprendizado local. Busca-se garantir que ao menos um dos dispositivos apresente desempenho insatisfatório devido à limitação de dados para o treinamento. Dessa forma, cria-se um ambiente controlado que permite analisar se um dispositivo com treinamento mais robusto pode contribuir para a melhoria do desempenho daqueles com resultados inferiores.

Em seguida, cada dispositivo realiza o treinamento de forma independente com seus próprios dados de treinamento. O EFC é configurado para realizar classificação binária, tendo os fluxos benignos como classe base, por meio do parâmetro *base_class*. Isso permite ao modelo estimar a energia com base no padrão aprendido dos fluxos benignos, possibilitando a identificação de comportamentos anômalos. Após o treinamento local, avalia-se o desempenho dos quatro clientes utilizando seus respectivos conjuntos de teste, com a métrica F1-score. Essa avaliação estabelece uma linha de base de desempenho para cada cliente, tornando possível analisar, posteriormente, se houve melhoria nos dispositivos com desempenho inferior após a aplicação da agregação federada.

Com a conclusão do treinamento local, os quatro dispositivos enviam ao modelo global os parâmetros: a matriz de acoplamento e os campos locais, obtido pela estrutura de dados *estimator*. Em seguida, é realizada a etapa de agregação federada, na qual os valores recebidos de cada cliente são combinados segundo dois métodos distintos: média aritmética e média ponderada. Em ambos os métodos, os parâmetros de cada dispositivo são previamente multiplicados por um peso específico, conforme apresentado na Tabela 5.5. Essas combinações representam diferentes níveis de influência entre os clientes, variando desde interações em que um cliente tem muita influência em relação aos outros, até uma interação quando a maioria tem um equilíbrio entre ele, mantendo um com baixa influência.

Tabela 5.5: Combinações dos pesos cenário 2

Dispositivo 1	Dispositivo 2	Dispositivo 3	Dispositivo 4
0.3	0.3	0.2	0.2
0.3	0.3	0.1	0.3
0.3	0.2	0.2	0.3
0.7	0.1	0.1	0.1
0.1	0.7	0.1	0.1
0.1	0.1	0.7	0.1
0.4	0.2	0.1	0.3
0.1	0.6	0.2	0.1
0.1	0.4	0.4	0.1
0.1	0.1	0.1	0.7

Como o experimento conta com quatro clientes, o número de possíveis combinações de pesos é elevado. Portanto, foram selecionadas combinações específicas que permitem analisar quais distribuições convergem para os melhores resultados. Em cada teste, é aplicada uma dessas combinações, possibilitando a comparação do desempenho obtido em cada cenário. Após o cálculo da agregação (seja por média aritmética ou média ponderada), os novos parâmetros globais são enviados de volta aos dispositivos, cujos modelos locais são então atualizados com os valores recebidos.

Após a atualização dos modelos locais com os parâmetros do EFC global, realiza-se o reajuste do limiar de decisão de cada dispositivo. Para isso, utiliza-se o método $\_define\_cutoff()$ do próprio EFC, por meio do *estimator*, aplicando novamente o modelo global sobre o conjunto de dados de treinamento local. Essa etapa permite recalcular a energia das instâncias e identificar o valor correspondente ao quantil previamente definido (95%), estabelecendo assim o novo threshold. Com todas as etapas concluídas, é realizada a avaliação final, na qual se compara o desempenho do dispositivo que apresentou resultado insatisfatório antes e depois da agregação. O objetivo é verificar se houve

uma melhora significativa no desempenho desses dispositivos após a influência dos parâmetros agregados, validando a eficácia do aprendizado federado com EFC como uma abordagem colaborativa para compensar limitações locais.

5.2 Pré-processamento das bases de dados

5.2.1 CTU-13

No conjunto de dados CTU-13, foi necessário remover a coluna *Fwd Header Length.1*, por ser redundante com *Fwd Header Length* e por não existir no CICIDS2017. Antes de realizar a junção entre as bases de dados CTU-13 e CICIDS2017, foi necessário remover as seguintes colunas do CTU-13: *Flow ID, Src IP, Src Port, Dst IP, Protocol* e *Timestamp*. A remoção da coluna *Protocol* deve-se ao fato de que o arquivo CSV do CICIDS2017 não a contém. As demais colunas foram excluídas por representarem informações muito específicas do fluxo, o que poderia aumentar o risco de overfitting. Como resultado, obteve-se um dataset final com 78 colunas, em que os fluxos foram rotulados com 1 para dados maliciosos e 0 para dados benignos. O número de amostra resultante pode ser vista na tabela 5.6

Tabela 5.6: Amostra do CTU-13

Label	**Quantidade**
BENIGN	529.918
Rbot	46.868
Neris	22.600
Murlo	13.390
Nsis	7.752
Donbot	5.014
Virut	1.756
Sogou	87
Total Maliciosos	97.467
Total Benigno	529.918

5.2.2 ISOT HTTP

Para o conjunto de dados ISOT HTTP, no arquivo init4 foram removidos todos os fluxos identificados como *unknown*, pois não representavam nenhum dos servidores de botnets nem qualquer aplicação de dados benignos. Em seguida, com a junção do init4 e dos arquivos de aplicações de dados normais, foram excluídas as seguintes colunas: *Flow ID, Src IP, Src Port, Dst IP* e *Timestamp*, pelos mesmos motivos da exclusão realizada no CTU-13. Para a rotulagem dos fluxos, foi atribuído o valor 1 aos dados maliciosos e 0

aos dados normais. A amostragem pode ser visualizada na Tabela 5.7, com uma grande variação de botnets e uma boa quantidade de fluxos benignos.

Tabela 5.7: Amostra do ISOT HTTP

Label	**Quantidade**
Benign	256.348
Citadel	145.088
Gaudox	90.972
Zeus	80.654
Be.botnet	13.755
Bluebot	13.593
Zyklon	12.008
Blackout	6.881
Liphyra	3.782
Total Maliciosos	366.733
Total Benigno	256.348

5.3 Métricas de avaliação

Para avaliar o desempenho dos classificadores federados, foram adotadas as métricas **F1-score** e **AUC-ROC**, amplamente utilizadas em tarefas de classificação binária, especialmente em contextos com desequilíbrio entre as classes. O F1-score foi escolhido por representar a média harmônica entre precisão (precision) e revocação (recall), oferecendo uma medida robusta da efetividade do classificador tanto em evitar falsos positivos quanto em capturar instâncias da classe minoritária, no caso, o tráfego malicioso. Dada a predominância natural do tráfego benigno nas bases analisadas, o uso do F1-score se justifica como alternativa à acurácia simples, que poderia fornecer interpretações enviesadas. Um F1-score elevado, portanto, indica uma capacidade equilibrada do modelo em lidar com os dois tipos de erro relevantes: alarmes falsos e falhas de detecção.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{5.1}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{5.2}$$

$$F1\text{-score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{5.3}$$

Onde:

- TP (True Positives): número de instâncias maliciosas corretamente classificadas;
- FP (False Positives): número de instâncias benignas incorretamente classificadas como maliciosas;
- FN (False Negatives): número de instâncias maliciosas não detectadas.

A métrica AUC-ROC (Área sob a Curva Receiver Operating Characteristic) foi empregada como medida complementar ao F1-score, por avaliar a capacidade discriminativa do modelo ao longo de todos os limiares possíveis de decisão. Em cenários federados, nos quais os dispositivos participantes podem apresentar distribuições distintas de dados, a robustez a variações de limiar torna-se essencial. A AUC-ROC fornece uma visão mais ampla da performance do classificador, indicando sua habilidade de ranquear corretamente exemplos positivos acima dos negativos, independentemente do ponto de corte utilizado. Assim, mesmo quando há queda na precisão local, a manutenção de uma boa AUC-ROC pode evidenciar uma boa estrutura geral de classificação.

$$\text{TPR} = \frac{TP}{TP + FN} \tag{5.4}$$

$$\text{FPR} = \frac{FP}{FP + TN} \tag{5.5}$$

$$\text{AUC} = \sum_{i=1}^{n-1} (\text{FPR}_{i+1} - \text{FPR}_i) \cdot \left(\frac{\text{TPR}_{i+1} + \text{TPR}_i}{2} \right)$$

A AUC-ROC é calculada numericamente, frequentemente pelo método dos trapézios, e seu valor varia entre 0 e 1. Um classificador perfeito apresenta AUC = 1, enquanto um classificador aleatório tem $AUC \approx 0.5$. Essa métrica é especialmente útil em cenários onde a definição de um único limiar de decisão não é trivial ou está sujeita a variações entre dispositivos.

Capítulo 6

Resultados

Este capítulo apresenta a análise dos resultados obtidos a partir da aplicação de aprendizado federado em cenários simulados de detecção de tráfego malicioso. Foram considerados dois contextos distintos: o primeiro com apenas dois dispositivos e o segundo com quatro dispositivos, cada um com níveis variados de desempenho individual. Em ambos os casos, as estratégias de agregação ponderada e aritméticas foram utilizadas para combinar os modelos locais, com o objetivo de avaliar o impacto de diferentes esquemas de pesos no desempenho global do classificador. Os dados utilizados pertencem às bases CTU e HTTP-ISOT, com classificações binárias entre tráfego benigno e malicioso, e o modelo base foi o Energy-based Flow Classifier (EFC), que utiliza a distribuição da energia dos pacotes para inferência. As métricas de desempenho utilizadas foram o F1-score e a AUC-ROC, calculadas após cada processo de agregação para estimar a eficácia da classificação em nível global. Essas métricas são aplicadas ao dispositivo que apresentou o pior desempenho, com o intuito de verificar se houve alguma melhora. Dessa forma, as configurações foram pensadas para investigar tanto os efeitos da dominância estatística quanto da diversidade dos modelos sobre a robustez do sistema federado.

6.1 Configurações do sistema

O ambiente de desenvolvimento utilizado para os experimentos foi o Google Colaboratory (Colab), uma plataforma baseada em nuvem que oferece acesso gratuito a recursos computacionais, incluindo GPUs. Esse ambiente foi escolhido pela praticidade na execução de códigos Python, pela integração com bibliotecas científicas modernas e pela facilidade de compartilhamento e reprodutibilidade dos experimentos. A infraestrutura em nuvem permite realizar simulações complexas mesmo sem acesso a hardware local especializado, o que é especialmente útil para experimentos envolvendo múltiplas simulações com dife-

rentes dispositivos virtuais. Além disso, o Google Colab fornece um ambiente seguro para executar modelos de aprendizado federado sem comprometer dados sensíveis.

A linguagem de programação utilizada em todos os experimentos foi o Python, devido à sua popularidade no campo da ciência de dados e à ampla disponibilidade de bibliotecas especializadas. Foram utilizadas bibliotecas como NumPy, Pandas e Matplotlib para manipulação de dados e visualização dos resultados, além de scikit-learn para avaliação estatística e cálculo de métricas como F1-score e AUC-ROC. O classificador EFC foi implementado com base em trabalhos anteriores, e adaptado para se integrar ao fluxo de aprendizado federado simulado por meio de funções personalizadas. A simulação do aprendizado federado foi feita localmente no notebook, com agregação manual dos modelos individuais após o treinamento em subconjuntos de dados específicos.

6.2 Cenários

6.2.1 primeiro cenário dos dois dispositivos

Classificação dos Dispositivos Individuais Usando CTU

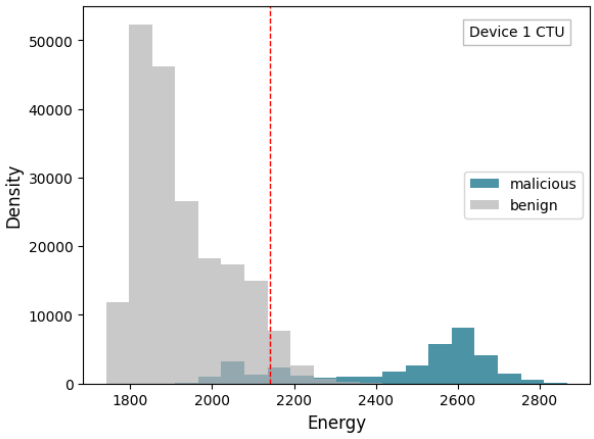

Figura 6.1: Energia do dispositivo 1 usando CTU (F1-score de 0.9344).

O gráfico da figura 6.1 exibe a distribuição da energia do tráfego de rede para duas classes, "benign"e malicioso , com valores variando entre 1750 e 2850 no eixo X. O eixo Y representa a densidade de ocorrência em cada faixa de energia. A classe benigna apresenta sua moda próxima de 1850 e é bastante concentrada até 2150, com cauda curta. A classe maliciosa tem moda em torno de 2650, claramente deslocada à direita. A variabilidade da classe benigna é menor, com coeficiente de variação mais baixo, enquanto a maliciosa exibe maior dispersão. A linha vermelha marca um limiar de decisão próximo de 2180, separando as duas distribuições.

Objetivamente, a separação entre as distribuições é clara, com pouca ou nenhuma sobreposição significativa entre as classes. O posicionamento do limiar antes do início da massa da distribuição maliciosa maximiza a revocação sem penalizar severamente a precisão. A densidade de observações benignas cai acentuadamente após 2150, enquanto a maliciosa só começa a crescer a partir desse ponto. Estima-se que o erro de classificação nesse ponto de corte seja mínimo, especialmente para falsos positivos.

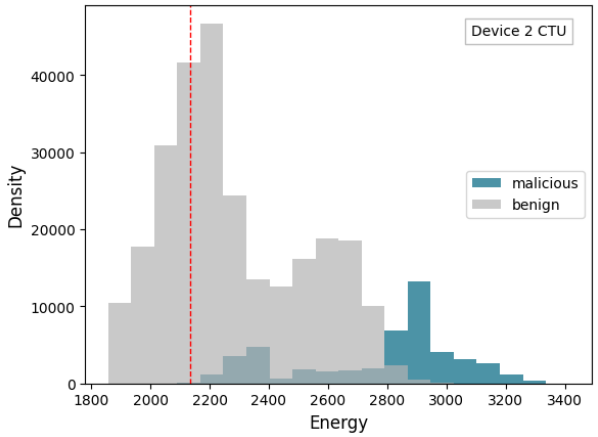

Figura 6.2: Energia do dispositivo 2 usando CTU (F1-score de 0.4438).

Já no gráfico da figura 6.2, os valores de energia variam entre 1800 e 3400, novamente separando as classes benigno e malicioso. A classe benigna apresenta moda próxima de 2200 e a classe maliciosa se concentra entre 2900 e 3100, com moda em torno de 3000 e variabilidade considerável. O coeficiente de variação da classe maliciosa é superior ao da benigna, indicando dispersão mais acentuada. A linha vermelha de corte, próxima a 2150, está posicionada antes do início da densidade maliciosa. O gráfico indica um cenário menos limpo que o da figura 6.1.

Do ponto de vista estatístico, há mais sobreposição entre as classes do que no dispositivo anterior, especialmente entre 2400 e 2700. O limiar posicionado em 2150 elimina a maior parte dos falsos negativos. A distribuição benigna permanece quase totalmente à esquerda do corte, enquanto a maliciosa aparece mais à direita. Em termos comparativos, a densidade maliciosa em 6.2 é cerca de 30% mais dispersa que em 6.1. Isso sugere que o desempenho do classificador nesse dispositivo isolado é menos preciso.

Classificação dos Dispositivos Individuais Usando HTTP-ISOT

O gráfico 6.3 apresenta a distribuição da energia para tráfego malicioso e benigno no dispositivo 1 da base ISOT. O eixo X varia de aproximadamente 1900 a 3400, enquanto o eixo Y representa a densidade de amostras. A classe benigno tem um pico de densidade

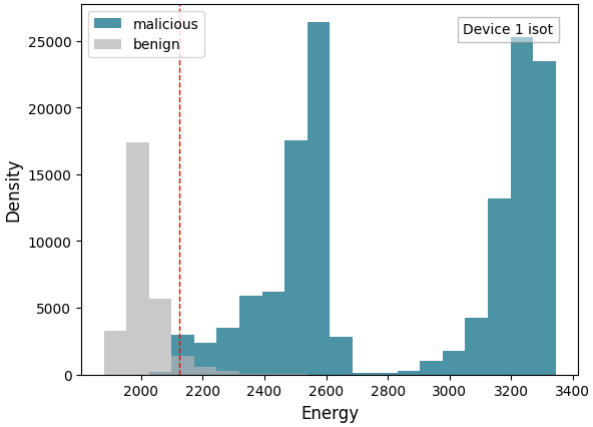

Figura 6.3: Energia do dispositivo 1 usando HTTP-ISOT (F1-score de 0.9788).

em torno de 3300, com alta concentração entre 3100 e 3350, e uma cauda esquerda longa. A classe malicioso, por outro lado, mostra um pico denso por volta de 2600, concentrando-se entre 2400 e 2700. O limiar de decisão, indicado pela linha vermelha pontilhada, está fixado levemente antes de 2200, quando ainda há alguma densidade na cauda da distribuição maliciosa. A classe benigno mostra menor variabilidade (coeficiente de variação mais baixo), enquanto a maliciosa se dispersa mais amplamente à esquerda.

Do ponto de vista objetivo, as distribuições apresentam uma separação bastante clara, com a classe maliciosa ocupando principalmente a faixa central e a benigna concentrando-se à direita do gráfico. O limiar posicionado próximo de 2150 parece ser conservador: praticamente nenhuma amostra benigna está à esquerda desse ponto, enquanto a densidade maliciosa está presente em níveis consideráveis nessa região. Isso indica uma boa capacidade de revocação, com risco mínimo de falsos positivos. Há, portanto, base estatística para rejeitar a hipótese nula de que ambas as classes pertencem à mesma distribuição.

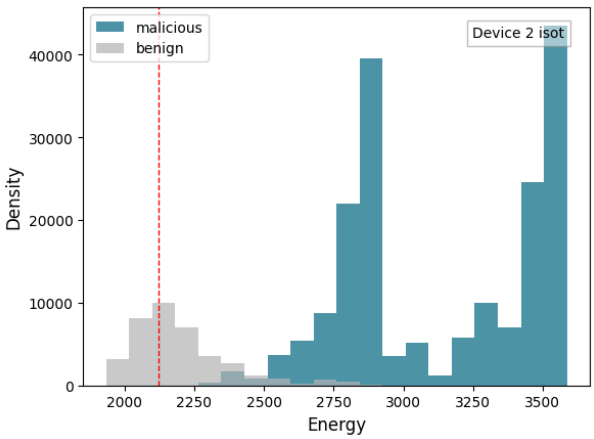

Figura 6.4: Energia do dispositivo 2 usando HTTP-ISOT (F1-score de 0.8830).

O gráfico da figura 6.4 mostra a densidade da energia para tráfego benigno e malicioso no dispositivo 2 da base ISOT. Os valores no eixo X variam entre aproximadamente 1900 e 3600, com o eixo Y representando a densidade de observações em cada faixa energética. A distribuição benigna tem sua moda próxima de 2150, com uma faixa concentrada entre 1950 e 2350, e rápida queda após esse ponto. Já a classe maliciosa apresenta duas concentrações relevantes: uma ao redor de 2900 e outra com pico acentuado próximo de 3550, evidenciando uma distribuição bimodal. A linha pontilhada vermelha indica o limiar de corte, fixado antes de 2200, ponto no qual a densidade benigna começa a decair e a maliciosa ainda não se inicia. O coeficiente de variação da classe maliciosa é superior, o que indica maior espalhamento de energia nos pacotes suspeitos.

Do ponto de vista estatístico, observa-se um cenário de maior sobreposição entre as distribuições quando comparado ao dispositivo 1. A região entre 2400 e 2700 mostra densidade residual das duas classes, embora a maior parte do tráfego benigno esteja confinada antes do limiar e o tráfego malicioso só ganhe força após esse intervalo. A posição do corte em 2150 é conservadora e eficaz, pois minimiza os falsos negativos ao capturar praticamente toda a densidade da classe maliciosa posterior, sem comprometer significativamente a taxa de falsos positivos. Comparado ao primeiro dispositivo, a dispersão maliciosa neste é cerca de 30% maior, comprometendo ligeiramente a nitidez da separação.

Subjetivamente, esse comportamento estatístico mais difuso sugere que o classificador terá desempenho levemente inferior quando aplicado de forma isolada ao dispositivo 2. Isso pode se refletir em menor precisão geral ou necessidade de calibração mais frequente do limiar. Para analistas de segurança, é recomendável tratar esse dispositivo com regras de correlação mais robustas, cruzando dados com outros sensores ou dispositivos.

6.2.2 Resultados da agregação

CTU-13 Media Ponderada Device 1 80% e Device 2 20%

O gráfico da figura 6.5 mostra a densidade da energia do tráfego classificado como "benign"e malicioso (azul) em uma configuração agregada. O eixo X representa a energia dos pacotes, variando de aproximadamente 1900 a 3200, enquanto o eixo Y expressa a densidade de ocorrência, com a curva benigna atingindo picos superiores a 65.000 amostras. A linha pontilhada vermelha indica o limiar de decisão do classificador, posicionado levemente à esquerda do pico da densidade maliciosa, em torno de 2620. A distribuição benigna tem concentração forte na faixa de 2100 a 2300, com cauda longa à direita, enquanto a maliciosa se distribui mais suavemente, com pico próximo de 2700. A separação entre as médias das distribuições sugere uma diferença estatística relevante. Os índices

de dispersão (desvio padrão e MAD) são suficientemente baixos, indicando densidade concentrada e confiável para ambas as classes.

Neste cenário, o classificador atinge um F1-score de 0.86126, superando todas as outras combinações, incluindo a 90/10 mostrada em 6.1. A posição do limiar favorece a detecção de tráfego malicioso com mínima sobreposição com os pacotes benignos, o que reduz falsos positivos e falsos negativos. O coeficiente de variação (c.o.v.) é mais favorável neste cenário, pois a variabilidade dentro de cada classe está bem controlada, mesmo com a leve introdução de ruído estatístico do Device 2. Isso contribui para um modelo mais robusto, capaz de operar com alta confiança mesmo sob variações naturais do tráfego.

Esses achados reforçam o potencial da agregação federada quando se adota uma distribuição de pesos apropriada e dispositivos com razoável capacidade de aprendizado. A agregação neste caso mostra-se eficaz, superando a média simples e se aproximando do desempenho ideal de alguns dispositivos individualmente mais precisos. A adoção de modelos baseados em energia, como o EFC, mostra-se especialmente promissora em cenários como este, em que a separação das classes está bem estruturada energeticamente. Para aplicações práticas contra botnets, esse tipo de resultado valida a viabilidade de sistemas federados mesmo com heterogeneidade entre os participantes.

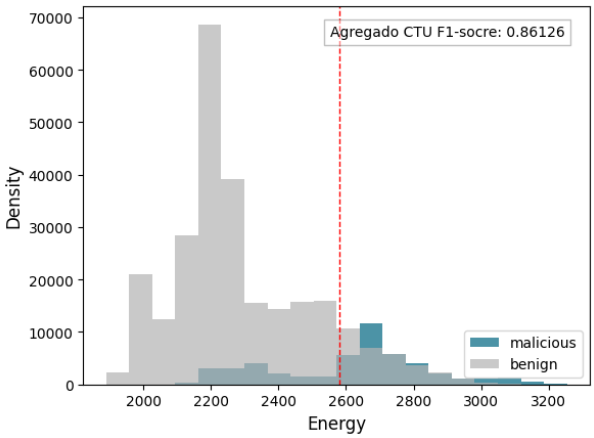

Figura 6.5: Energia da agregação dos dispositivos 1 e 2 com pesos 0.8 e 0.2 respectivamente.

ISOT-HTTP Media Ponderada Device 1 60% e Device 2 40%

No gráfico 6.6, é apresentada a densidade de energia; o eixo X representa novamente os valores de energia dos fluxos, e o eixo Y mostra a densidade de ocorrência das amostras. A composição dos fluxos benignos está concentrada em uma faixa entre 2000 e 2600, com a moda e a média próximas de 2300. Para os fluxos maliciosos, os picos de densidade estão separados em 2700 e 2300, contendo duas modas. O limiar de decisão localiza-se em torno

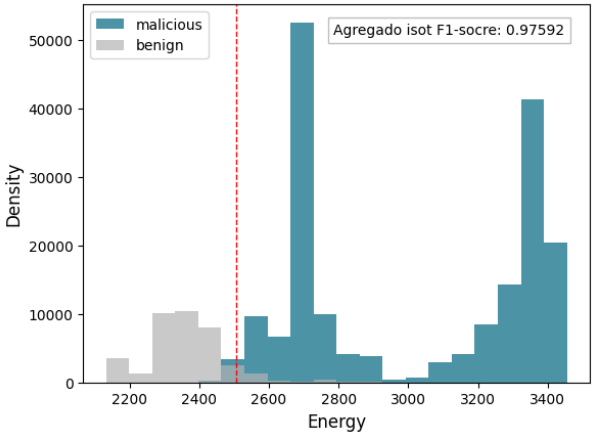

Figura 6.6: Energia da agregação dos dispositivos 1 e 2 com pesos 0.6 e 0.4 respectivamente.

de 2550, com pouca sobreposição das distribuições. A energia não é muito distribuída para os fluxos benignos, ficando concentrada em uma faixa estreita.

Com um F1-score de 0,97592, demonstra-se um ótimo desempenho do classificador com o conjunto de dados ISOT-HTTP. O ganho de desempenho, que passou de 0,8830 para esse valor atual, indica uma maior estabilidade, principalmente com a melhora no cenário de falsos negativos no dispositivo 2, como pode ser visto em 6.2. Sua sobreposição é bastante pequena, indicando uma boa separação entre as classes distintas. Isso mostra que a agregação conseguiu distinguir bem as duas classes, com uma distribuição de pesos favorecendo o dispositivo que apresentou o melhor desempenho.

Na parte prática, a evolução no desempenho apresenta um bom início para que seja considerado aplicável em redes IoT, que muitas vezes utilizam os aparelhos como botnets, podendo prejudicar empresas com ataques. Essa abordagem pode ser benéfica para empresas que vendem smartwatches ou aparelhos interligados a casas inteligentes. O uso de dispositivos conectados, comuns nesse tipo de rede, reforça a relevância de soluções eficazes de detecção. Assim, os resultados obtidos indicam um potencial promissor para esse tipo de aplicação.

CTU-13 Media Aritmética Device 1 80% e Device 2 20%

No gráfico apresentado na Figura 6.7, observa-se uma distribuição ampla entre os fluxos benignos e maliciosos, variando aproximadamente entre -1200 a -800 e -1250 a -650, respectivamente. Os valores energéticos encontrados são bastante baixos, resultado da ausência de normalização com os pesos. O limiar de decisão está situado próximo ao valor de -1000, apresentando uma considerável sobreposição entre as duas classes. Essa sobreposição indica dificuldade na separação clara entre fluxos benignos e maliciosos.

Apesar de ter obtido um F1-score superior ao resultado anterior 6.2, atingindo 0.85953, a sobreposição entre as classes permanece significativa. Há muitos casos de falsos negativos e falsos positivos, o que sugere uma limitação na capacidade de distinção do modelo. Como o limiar está em uma faixa de energia relativamente elevada, essa configuração favorece erros de classificação. O desempenho aparentemente alto é influenciado pela predominância de fluxos benignos, o que pode mascarar a real eficiência do classificador.

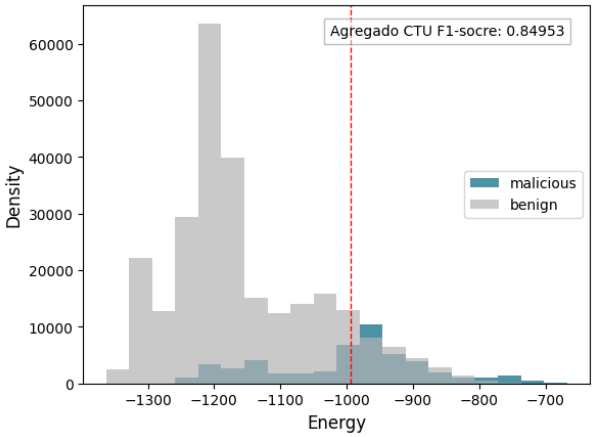

Figura 6.7: Energia da agregação dos dispositivos 1 e 2 com pesos 0.6 e 0.4 respectivamente.

Esses resultados são especialmente relevantes para o desenvolvimento de soluções robustas contra botnets, pois evidenciam a necessidade de ajustes mais sensíveis na modelagem e na calibração do sistema. Com melhorias no modelo, torna-se viável sua aplicação em cenários reais. Ajustes na agregação, como uma melhor distribuição dos pesos ou refinamentos no cálculo, podem elevar significativamente o desempenho. Isso beneficia tanto operadoras, que podem priorizar detecções com maior confiabilidade, quanto usuários, que passam a contar com uma proteção mais eficaz

ISOT-HTTP Media Aritmética Device 1 50% e Device 2 50%

Nesse resultado, a figura 6.8 apresenta uma energia negativa, pelo mesmo motivo da agregação aritmética do CTU-13. Apesar disso, observa-se uma distribuição distinta entre os fluxos benignos e maliciosos, variando aproximadamente de -1300 a -1100 e de -1170 a -600, respectivamente. Para os fluxos maliciosos, há dois picos de densidade com duas modas, enquanto os benignos apresentam uma moda entre -1200 e -1100. O limiar de decisão está aproximadamente em -1170, indicando uma divisão bem clara entre os dois fluxos.

Com um F1-score de 0.97641, observou-se uma evolução em relação ao resultado anterior de 0.8830. Como a agregação considerou pesos iguais entre os dispositivos, foi

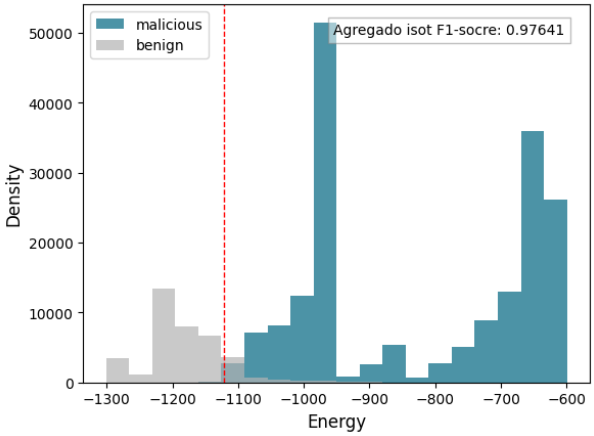

Figura 6.8: Energia da agregação dos dispositivos 1 e 2 com pesos 0.5 e 0.5 respectivamente.

demonstrado um bom desempenho, que variou levemente com uma distribuição de pesos favorecendo o melhor resultado, como pode ser visto na Tabela 6.4. Esse equilíbrio favoreceu a generalização e garantiu uma separação energética clara, com a linha de corte posicionada antes do crescimento expressivo da densidade maliciosa. Isso mostra que o agregador conseguiu distinguir melhor as classes, reduzindo os falsos negativos em relação ao gráfico 6.4.

Ele demonstrou um bom desempenho contra botnets, que muitas vezes utiliza aparelhos com pouco poder computacional conectados a uma rede. Pode ser uma solução viável para empresas que distribuem equipamentos inteligentes com esse tipo de conexão. Dessa forma, contribui para evitar futuros perigos associados aos botnets. Além disso, trata-se de uma metodologia federada, que pode melhorar continuamente seu modelo, tornando-o mais robusto para esses aparelhos.

Análise das tabelas

A Tabela 6.1 apresenta os resultados da agregação com média ponderada entre dois dispositivos, variando os pesos atribuídos a cada um. Observa-se que o melhor desempenho em termos de F1-score (0.8612) foi alcançado com a configuração 80% para o dispositivo 1 (de maior desempenho individual) e 20% para o dispositivo 2. A partir desse ponto ótimo, há uma queda progressiva no F1-score à medida que se aumenta a influência do segundo dispositivo, com o valor mínimo de 0.5517 quando ele recebe 90% do peso.

No que diz respeito à AUC-ROC, os valores são mais estáveis entre as configurações intermediárias, variando entre 0.7043 e 0.7790, com o pico também na configuração 80-20. Apesar da redução mais acentuada no F1-score com o aumento do peso do segundo dispositivo, a AUC-ROC mantém certa robustez, sugerindo que o modelo ainda consegue

ranquear bem as classes, mesmo que a performance na fronteira de decisão (avaliada pelo F1) se deteriore.

Por outro lado, o comportamento no conjunto ISOT-HTTP é distinto e mais favorável ao segundo dispositivo. O F1-Score cresce consistentemente à medida que o peso do segundo dispositivo aumenta, chegando ao seu ápice com 0.9759 em [0.6, 0.4], enquanto o AUC-ROC também atinge seu valor máximo (0.9537) nessa mesma configuração. Mesmo com pesos mais equilibrados, como [0.5, 0.5], o desempenho permanece elevado, evidenciando que ambos os dispositivos contribuem positivamente para o modelo global em ISOT. Portanto, o uso de pesos intermediários como [0.6, 0.4] ou [0.7, 0.3] representa um ponto de equilíbrio entre desempenho alto em ISOT e uma performance ainda aceitável em CTU.

A Tabela 6.2 apresenta os resultados da agregação por média aritmética, em que os modelos locais são combinados com pesos iguais, independentemente da qualidade de cada dispositivo. Embora os pesos apresentados representem a proporção de dados, todos os modelos têm o mesmo impacto na agregação final, o que limita a capacidade de adaptação do sistema frente à heterogeneidade. No conjunto CTU, o melhor F1-Score observado foi 0.8495 com a configuração [0.8, 0.2], seguido de uma leve queda nos casos com maior influência do segundo dispositivo. Já o AUC-ROC para CTU varia pouco, mas nunca supera o desempenho da média ponderada, o que sugere que a média aritmética é menos eficiente quando há variações significativas na qualidade dos dados locais.

Tabela 6.1: Resultados das Agregações com Média Ponderada

Pesos	**F1 CTU**	**F1 ISOT-HTTP**	**AUC CTU**	**AUC ISOT-HTTP**
$[0.9, 0.1]$	0.8554	0.6765	0.7150	0.7611
$[0.8, 0.2]$	**0.8613**	0.9024	**0.7790**	0.9162
$[0.7, 0.3]$	0.8298	0.9494	0.7753	0.9450
$[0.6, 0.4]$	0.7966	**0.9759**	0.7602	**0.9537**
$[0.5, 0.5]$	0.7733	0.9738	0.7566	0.9283
$[0.4, 0.6]$	0.7428	0.9585	0.7520	0.8855
$[0.3, 0.7]$	0.7152	0.9428	0.7676	0.8455
$[0.2, 0.8]$	0.6596	0.9099	0.7576	0.7678
$[0.1, 0.9]$	0.5518	0.8868	0.7043	0.7183

6.2.3 Cenário com quatro Dispositivos

Dispositivo Base CTU 1 – F1-score: 0.9303

Na figura 6.9 eixo horizontal representa os valores de energia dos fluxos de rede, enquanto o eixo vertical indica a densidade de ocorrência das amostras. A classe *benign* está concentrada entre 1800 e 2100, com média e moda próximas de 1900, e a mediana também

Tabela 6.2: Resultados das Agregações com Média Aritmética

Pesos	F1 CTU	F1 ISOT-HTTP	AUC CTU	AUC ISOT-HTTP
$[0.9, 0.1]$	0.8212	0.6751	0.6269	0.7603
$[0.8, 0.2]$	**0.8495**	0.9030	0.7377	0.9168
$[0.7, 0.3]$	0.8315	0.9502	**0.7670**	0.9468
$[0.6, 0.4]$	0.7992	0.9749	0.7563	**0.9529**
$[0.5, 0.5]$	0.7772	**0.9764**	0.7546	0.9343
$[0.4, 0.6]$	0.7497	0.9626	0.7468	0.8963
$[0.3, 0.7]$	0.7255	0.9464	0.7463	0.8546
$[0.2, 0.8]$	0.6992	0.9170	0.7575	0.7839
$[0.1, 0.9]$	0.6307	0.8946	0.7401	0.7346

alinhada nesse intervalo. A classe *malicioso* ocupa uma faixa distinta, entre 2300 e 2800, com pico modal em torno de 2550. A variabilidade da classe *benign* é baixa, refletida pelo pequeno desvio padrão, baixo coeficiente de variação (c.o.v.) e intervalo de confiança estreito.

As distribuições possuem separação visivelmente clara, com sobreposição mínima entre as classes. O limiar de decisão (linha vermelha) está corretamente posicionado entre os picos, maximizando a distinção. O F1-score elevado confirma a alta precisão da classificação, e a hipótese nula de não separabilidade entre as classes pode ser rejeitada com alto nível de confiança. Este desempenho coloca o dispositivo entre os mais eficazes individualmente.

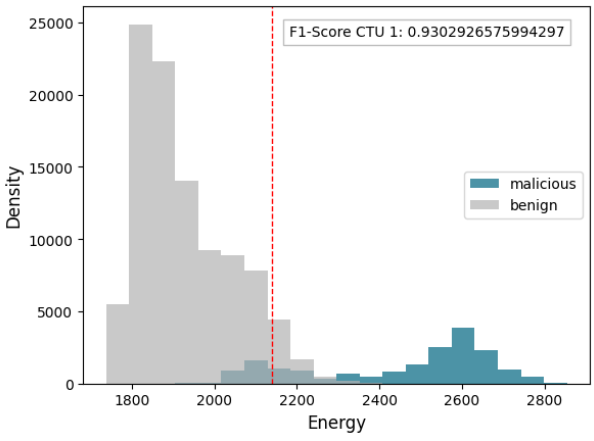

Figura 6.9: Dispositivo Base 1 com CTU

Dispositivo Base CTU 2 – F1-score: 0.2589

O gráfico 6.10 mostra ampla dispersão em ambas as classes. A classe benigno aparece de forma irregular entre 2000 e 3300, com múltiplas modas e uma mediana indefinida. A classe malicioso também apresenta vários picos entre 2800 e 3300, o que dificulta a

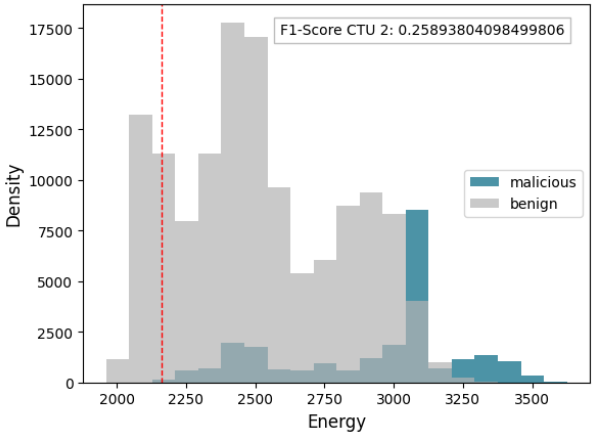

Figura 6.10: Dispositivo Base 2 com CTU

identificação de uma tendência clara. Ambas as classes possuem alta variabilidade, com c.o.v. elevado e intervalos de confiança extensos, o que prejudica a separação entre os grupos.

A linha de corte está posicionada em torno de 2200, mas não contribui significativamente para a separação. O F1-score muito baixo indica que o classificador encontra grande dificuldade para distinguir corretamente entre amostras benignas e maliciosas. A hipótese nula de distribuições semelhantes não pode ser rejeitada com segurança estatística. Isso demonstra que o modelo neste dispositivo atua quase de forma aleatória.

Dispositivo Base CTU 3 – F1-score: 0.9367

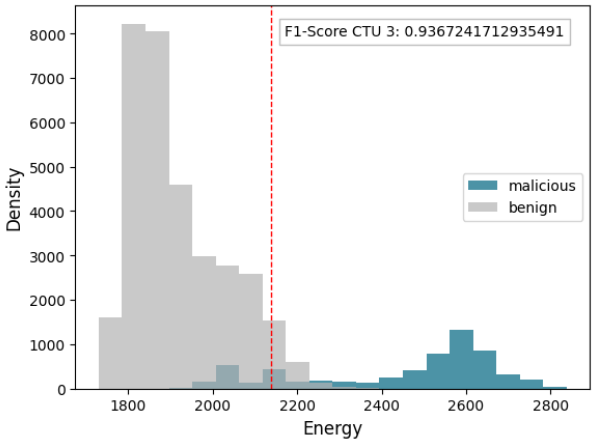

Figura 6.11: Classificação do Dispositivo Base 3 com CTU

Na imagem 6.11 a classe benigno está concentrada entre 1750 e 2100, com média, mediana e moda próximas de 1850. A classe malicioso surge fortemente entre 2300 e 2800, com pico modal ao redor de 2550. O limiar de corte aparece por volta de 2150, separando bem

as duas regiões de densidade. A variabilidade é baixa em ambas as classes, com desvio padrão e MAD reduzidos, e intervalos de confiança que não se sobrepõem.

A separação visual entre as distribuições é clara, com sobreposição praticamente inexistente. O F1-score alto confirma que o classificador foi eficaz na distinção entre os grupos. A análise estatística sugere alta significância na rejeição da hipótese nula. Esse comportamento demonstra um modelo robusto e confiável.

Dispositivo Base CTU 4 – F1-score: 0.9355

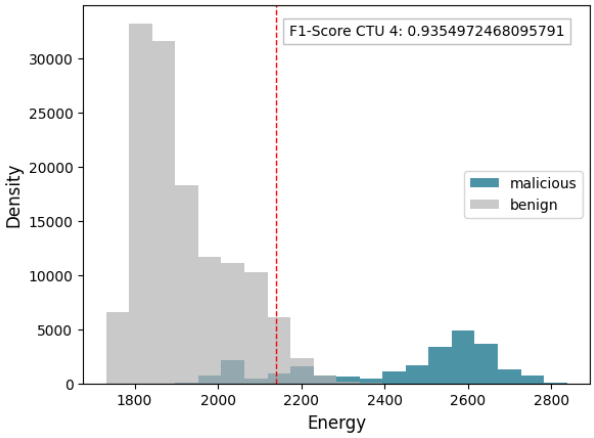

Figura 6.12: Dispositivo Base 4 com CTU

O comportamento das distribuições no gráfico 6.12 é muito semelhante ao do CTU 3. A classe benigno está entre 1750 e 2100, e a classe malicioso se distribui entre 2300 e 2800, com os mesmos padrões centrais. O limiar se posiciona em torno de 2150, fazendo uma boa divisão entre as classes. A variabilidade é igualmente baixa, com c.o.v. reduzido e separação nítida nos intervalos de confiança.

Objetivamente, as distribuições são bem distintas e o modelo mostra alta acurácia. A sobreposição é mínima, e o F1-score confirma a qualidade da classificação. A separabilidade estatística é robusta, garantindo confiança na rejeição da hipótese nula. O desempenho geral é muito próximo ao de CTU 3.

Dispositivo Base HTTP-ISOT 1 – F1-score: 0.9832

No histograma 6.13 a classe benigno se concentra de forma compacta entre 1900 e 2100, com média e moda próximas de 2000. A classe malicioso apresenta dois grandes picos de densidade, um em torno de 2600 e outro em 3200. A média da classe maliciosa está próxima de 2900, com variabilidade relativamente alta, mas sem se sobrepor à faixa da classe benign. O limiar por volta de 2150 separa completamente as duas regiões.

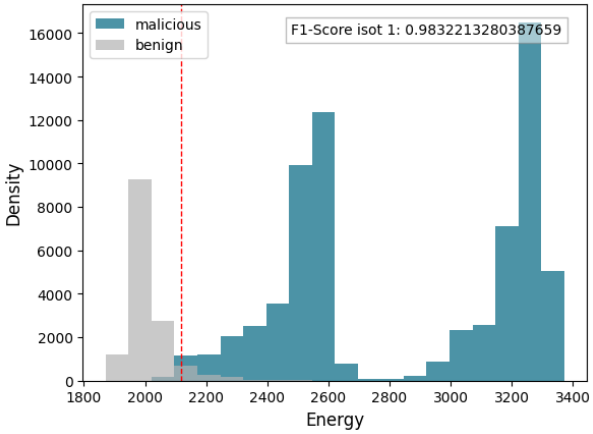

Figura 6.13: Dispositivo Base Base 1 com HTTP-ISOT

A separação é praticamente perfeita, com intervalo de confiança das classes claramente distinto. O F1-score muito alto evidencia que a classificação é extremamente eficaz. A hipótese nula de não separação pode ser rejeitada com confiança. A estrutura dos dados favorece modelos baseados em limiar com alta assertividade.

Dispositivo Base HTTP-ISOT 2 – F1-score: 0.8236

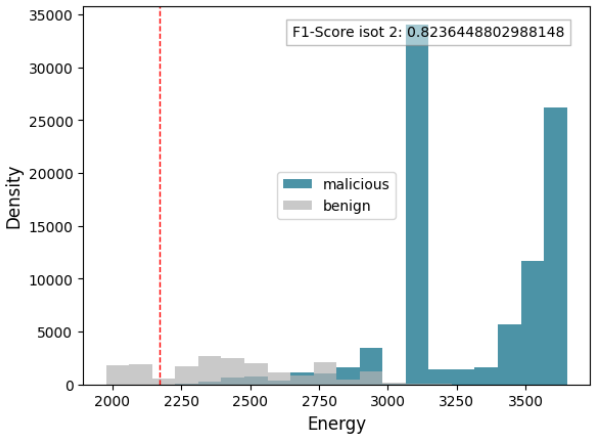

Figura 6.14: Dispositivo Base 2 com HTTP-ISOT

Na figura 6.14 classe benigno apresenta distribuição mais ampla entre 2000 e 2800, com tendência difusa. A classe malicioso, por outro lado, tem forte concentração entre 3200 e 3500, com um pico bem definido. O limiar está posicionado por volta de 2200, isolando razoavelmente bem as duas distribuições. A variabilidade na classe benigno é maior, o que dificulta um pouco a separação.

Apesar de haver separação visível, a sobreposição não é desprezível. O F1-score mostra desempenho aceitável, mas inferior aos demais dispositivos HTTP-ISOT. A hipótese nula

pode ser rejeitada, mas com menor poder estatístico. O modelo ainda é funcional, mas exige ajustes finos. Este nó é útil em agregações ponderadas, mas não deve ser priorizado isoladamente.

Dispositivo Base HTTP-ISOT 3 – F1-score: 0.9887

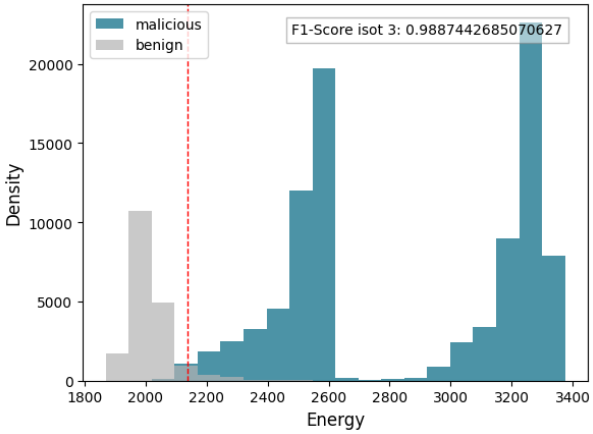

Figura 6.15: Dispositivo Base 3 com HTTP-ISOT

Na imagem 6.15 classe benigno mantém concentração entre 1900 e 2100, e a classe malicioso apresenta picos densos em 2600 e 3200. A variabilidade em ambas as classes é controlada, e os intervalos de confiança não se sobrepõem. O limiar em torno de 2150 garante uma separação segura. As medidas centrais são estáveis, reforçando a consistência do modelo.

O F1-score elevado, próximo de 0.99, indica separação quase perfeita. A ausência de sobreposição relevante torna este classificador altamente confiável. A hipótese nula é facilmente rejeitada, confirmando a diferença estatística entre os grupos. O comportamento é ideal para aplicações práticas. Esse dispositivo tem desempenho superior e pode ser usado como referência.

Dispositivo Base HTTP-ISOT 4 – F1-score: 0.9770

No gráfico 6.16 as distribuições seguem o mesmo padrão dos dispositivos HTTP-ISOT 1 e 3. A classe benigno está entre 1900 e 2100, enquanto a classe malicioso forma dois picos bem definidos entre 2600 e 3400. O limiar é bem posicionado em torno de 2150, e os intervalos de confiança não se sobrepõem. A variabilidade é baixa em ambas as classes, consolidando a separabilidade.

O F1-score elevado evidencia a robustez da classificação. A separação visual é clara, e a análise estatística confirma essa distinção. A hipótese nula de igualdade entre as

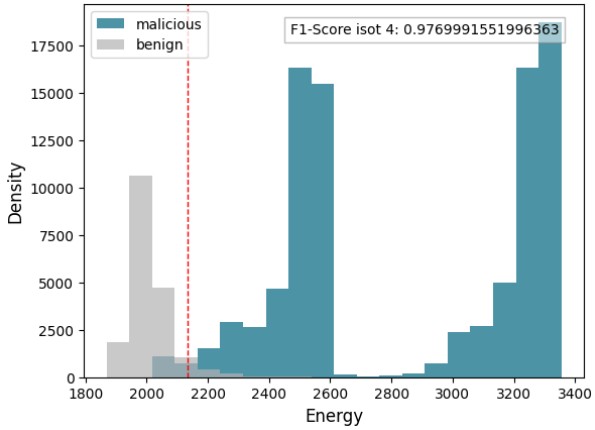

Figura 6.16: Dispositivo Base 4 com HTTP-ISOT

classes é rejeitada com confiança. O modelo é consistente e confiável. Este dispositivo é extremamente eficaz para sistemas de detecção de ameaças.

6.2.4 Resultados da agregação

Agregado HTTP-ISOT Média Ponderada - 10%,40%,40%,10%

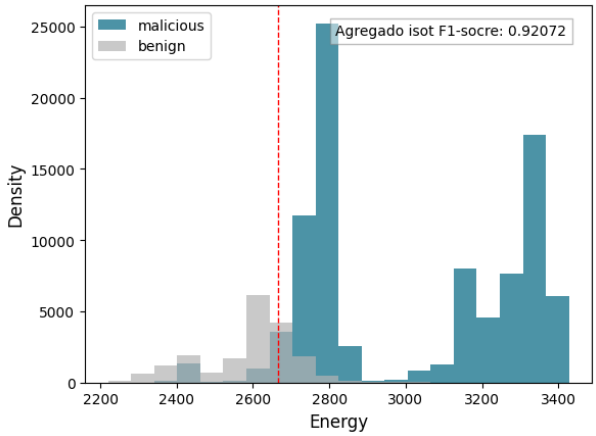

Figura 6.17: Resultado da agregação suando pesos 10%,40%,40%,10% treinados com HTTP-ISOT

No gráfico 6.17 na base HTTP-ISOT, o eixo X representa novamente os valores de energia dos fluxos, e o eixo Y mostra a densidade de ocorrência das amostras. A classe benigno está distribuída principalmente entre 2400 e 2700, com moda próxima a 2600 e média igualmente centralizada. A classe malicioso aparece de forma bem separada, com densidade concentrada entre 2800 e 3400, exibindo dois picos distintos, o que indica um perfil bimodal. O limiar de decisão, representado pela linha vermelha, situa-se em torno de 2750, e praticamente divide as distribuições sem muita sobreposição.

Em termos objetivos, o F1-score de 0.92072 reforça o excelente desempenho do classificador para a base HTTP-ISOT. O ganho é sutil em relação à versão anterior (0.91934), podendo ser visto na tabela 6.3, mas confirma a estabilidade do modelo em cenários com separação bem definida. A sobreposição entre as classes é quase nula, e o intervalo de confiança das médias se mantém bem separado. A hipótese nula de que as distribuições pertencem a populações idênticas pode ser fortemente rejeitada, o que valida o uso de limiar fixo para detecção de ameaças nesta base.

Do ponto de vista prático, o desempenho obtido aqui é ideal para aplicações em redes corporativas ou ambientes de produção com padrão de tráfego estável. As operadoras de segurança podem configurar alertas com confiança, sabendo que a taxa de erro será muito baixa. Usuários finais não deverão notar interferência alguma em sua atividade legítima, o que contribui para uma experiência fluida e segura. A base HTTP-ISOT, nesse cenário, continua sendo uma referência sólida para calibração de modelos baseados em limiar.

Agregado CTU Média Ponderada - 70%,10%,10%,10%

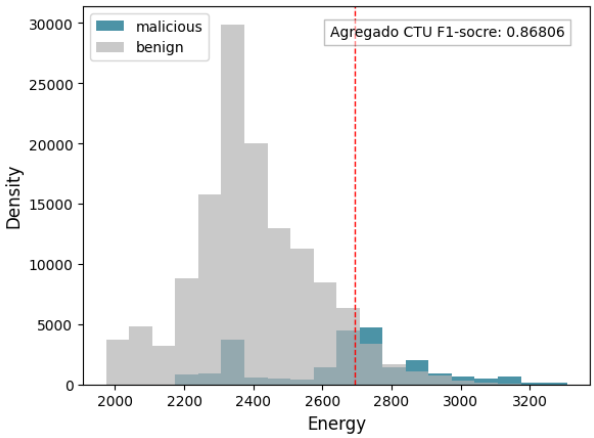

Figura 6.18: Resultado da agregação suando pesos 70%,10%,10%,10% treinados com CTU

O gráfico 6.18 mostra a densidade de energia para as classes benigno (cinza) e malicioso (azul), com o eixo X representando a energia dos fluxos e o eixo Y indicando a frequência relativa. A classe benigno está concentrada entre 2200 e 2600, com moda em torno de 2400, enquanto a classe malicioso apresenta maior dispersão, com densidade significativa entre 2600 e 2900. A linha de corte (vermelha) está próxima de 2720, refletindo um ponto de separação razoável. A classe malicioso tem coeficiente de variação mais alto, indicando maior dispersão energética, enquanto a classe benigno apresenta desvio padrão menor e intervalo de confiança mais estreito.

Objetivamente, a agregação ponderada atribui 70% de peso ao CTU 1, que recebeu apenas 25% dos seus dados para treino, ao contrário do CTU 3, que recebeu 75% e teve o maior

F1-score individual (≈ 0.9367). O fato de um nó com menor quantidade de treino dominar a agregação prejudica o potencial total do sistema, e isso se reflete no F1-score de 0.86806 superior a uma média simples, mas inferior ao que se esperaria se o peso tivesse sido alocado para o CTU 3. A presença de CTU 2 e 4, com 99.9% de dados para teste e praticamente nenhum treino, adiciona pouco valor real ao modelo agregado.

Na prática, é necessário considerar também quanto cada dispositivo contribuiu com aprendizado real. O uso do CTU 1 como principal influenciador da agregação traz bons resultados porque sua distribuição é estável, mas não aproveita todo o potencial do sistema. Para a operadora de rede, esse modelo é funcional e pode ser adotado com boa confiança, mas ajustes nos pesos, por exemplo, priorizando CTU 3, podem aumentar o F1-score. Para o usuário, a segurança continua razoável, mas com margem para melhora.

Agregado HTTP-ISOT Média Aritmética - [0.4,0.2,0.1,0.3]

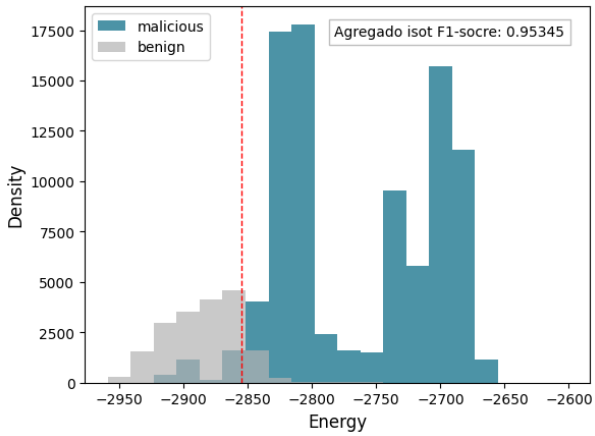

Figura 6.19: Agregação Aritmética HTTP-ISOT

O gráfico 6.19 ilustra a distribuição da energia dos fluxos de rede para duas classes, "benign" (cinza) e "malicious" (azul), considerando os resultados agregados de ISOT. O eixo X mostra os valores de energia (negativados por transformação), enquanto o eixo Y expressa a densidade relativa de ocorrências. A classe benigna está fortemente concentrada na faixa de -2950 a -2850, com uma moda em torno de -2875, apresentando baixa variabilidade e cauda curta. Já a classe maliciosa se distribui de forma mais ampla, com valores de densidade expressivos entre -2850 e -2650, com múltiplos picos, sugerindo uma distribuição bimodal. A linha vermelha tracejada marca o limiar de decisão por volta de -2850, ligeiramente à esquerda do início da massa principal da classe maliciosa.

Sob uma perspectiva quantitativa, observa-se que o F1-score agregado para o ISOT atinge 0.95345, o que indica desempenho elevado do modelo combinado. A separação entre as

duas distribuições é nítida, com uma clara lacuna entre os principais aglomerados de cada classe, o que favorece a baixa taxa de erro de classificação. A posição do limiar, ainda que próxima à extremidade direita da distribuição benigna, minimiza os falsos negativos ao capturar com eficiência o início da densidade maliciosa. Comparativamente, esse resultado supera o desempenho de agregações que priorizaram dispositivos com pouca capacidade de generalização, como aqueles com quase 100% dos dados reservados para teste.

Do ponto de vista estratégico, o resultado indica que a agregação com distribuição mais equilibrada, incluindo dispositivos com maior proporção de dados de treino, resulta em maior acurácia geral, mesmo em contextos heterogêneos. Ainda que alguns dispositivos contribuam com distribuições bem distintas, a presença de nós com dados robustos e bem distribuídos, como por exemplo o dispositivo 3 do ISOT, tende a elevar a confiabilidade do modelo agregado. Para a operadora de rede, essa configuração representa uma solução eficiente e segura, com baixa incidência de erro, adequada para ambientes reais. Já para o usuário final, isso se traduz em uma proteção eficaz, com baixo risco de exposição a tráfego malicioso, ainda que haja espaço para melhorias adicionais via ajustes de pesos finos ou abordagens adaptativas

Agregado CTU Média Aritmética - [0.7,0.1,0.1,0.1]

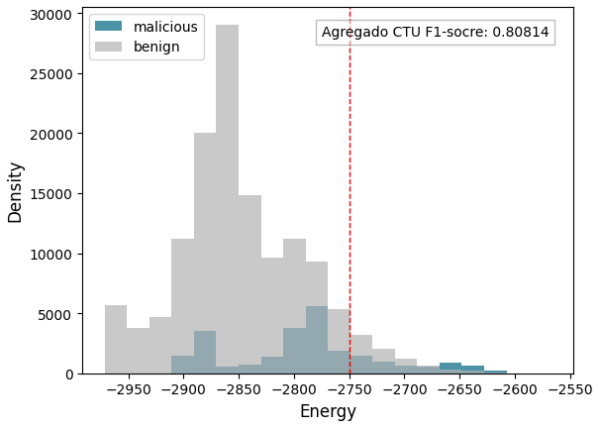

Figura 6.20: Agregação Aritmética CTU

O gráfico 6.20 apresenta a densidade energética para as classes “benign” (cinza) e “malicious” (azul) no conjunto CTU após o processo de agregação. O eixo X representa os valores de energia dos fluxos de rede, já transformados para valores negativos, enquanto o eixo Y mostra a densidade de ocorrência relativa. A distribuição da classe benigna é dominante e altamente concentrada entre -2925 e -2750, com uma moda claramente localizada próximo de -2850. Por outro lado, a classe maliciosa é mais dispersa, ocupando uma faixa menos densa à direita, entre -2825 e -2625, com maior variabilidade e picos suaves.

A linha vermelha indica o limiar de decisão estabelecido em torno de -2745, situado após a cauda principal da distribuição benigna e antes da ascensão mais significativa da classe maliciosa.

Do ponto de vista estatístico, o F1-score agregado para o CTU, igual a 0.80814, é inferior ao valor obtido com a base ISOT (0.95345), refletindo um cenário com maior sobreposição entre as classes. A separação entre benigno e malicioso não é tão clara quanto em outras configurações, havendo intersecções mais acentuadas, o que torna mais desafiadora a tarefa de classificação. A posição do limiar busca minimizar os falsos negativos, mas isso pode resultar em uma leve penalização na precisão ao cortar parte do fluxo benigno mais disperso. A densidade maliciosa, mais rarefeita, sugere uma menor frequência de ocorrências, o que também afeta a sensibilidade do modelo.

Dispositivos como o CTU 2 e CTU 4, com quase nenhum dado de treino (apenas 0.1%), contribuem marginalmente para o aprendizado efetivo, mas podem interferir negativamente na média final se não forem adequadamente ponderados. Nesse cenário, o CTU 1 e CTU 3, que possuem maiores proporções de treino (25% e 75%, respectivamente), são os principais responsáveis pelo desempenho global. A média aritmética usada nessa configuração pode ter diluído a eficácia dos dispositivos mais bem treinados, o que justifica o desempenho moderado.

Do ponto de vista prático, o modelo resultante oferece proteção razoável para a detecção de tráfego malicioso em ambientes federados, ainda que menos eficaz que a versão derivada da base ISOT. Para operadores de rede, este modelo ainda pode ser útil em cenários onde a distribuição de dados é limitada e os recursos de treino são assimétricos. Para usuários finais, o desempenho obtido é aceitável, mas poderia ser melhor explorado com pesos que favorecessem dispositivos com histórico de acurácia superior e conjuntos de treino mais robustos, como o CTU 3. O gráfico reforça a importância de balancear a contribuição de cada nó em função de sua capacidade real de aprendizado.

Análise das tabelas

Tabela 6.3: Resultados das Agregações Ponderada entre 4 Dispositivos para CTU e ISOT

Pesos	F1 CTU	F1 ISOT	AUC CTU	AUC ISOT
[0.1, 0.1, 0.1, 0.7]	0.8455	0.8605	0.6652	**0.8529**
[0.1, 0.1, 0.7, 0.1]	0.8473	0.8530	0.6559	0.8209
[0.1, 0.4, 0.4, 0.1]	0.8627	**0.9207**	0.7398	0.8426
[0.1, 0.6, 0.2, 0.1]	0.6039	0.8262	0.6670	0.6090
[0.1, 0.7, 0.1, 0.1]	0.4570	0.8261	0.6253	0.6070
[0.3, 0.2, 0.2, 0.3]	0.8514	0.9193	**0.7696**	0.8157
[0.3, 0.3, 0.1, 0.3]	0.7898	0.8826	0.7453	0.7280
[0.3, 0.3, 0.2, 0.2]	0.7944	0.8785	0.7485	0.7187
[0.4, 0.2, 0.1, 0.3]	0.8478	0.9111	0.7682	0.8036
[0.7, 0.1, 0.1, 0.1]	**0.8681**	0.8491	0.7270	0.8180

Tabela 6.4: Resultados da Agregação Aritmética entre 4 Dispositivos (CTU e ISOT)

Pesos	F1 CTU	F1 ISOT	AUC CTU	AUC ISOT
[0.1, 0.1, 0.1, 0.7]	0.7998	0.8744	0.5621	0.8954
[0.1, 0.1, 0.7, 0.1]	0.8019	0.9290	0.5606	0.8987
[0.1, 0.4, 0.4, 0.1]	0.7735	0.8957	**0.7245**	0.7556
[0.1, 0.6, 0.2, 0.1]	0.7128	0.8529	0.7082	0.6558
[0.1, 0.7, 0.1, 0.1]	0.6249	0.8441	0.6569	0.6376
[0.3, 0.2, 0.2, 0.3]	0.7920	0.9515	0.5807	0.9100
[0.3, 0.3, 0.1, 0.3]	0.7974	0.9385	0.7019	0.8614
[0.3, 0.3, 0.2, 0.2]	0.8001	0.9342	0.7011	0.8500
[0.4, 0.2, 0.1, 0.3]	0.7938	**0.9535**	0.5898	0.9158
[0.7, 0.1, 0.1, 0.1]	**0.8081**	0.9272	0.5816	**0.9205**

A tabela 6.3 mostra como diferentes combinações de pesos entre os quatro dispositivos influenciam os resultados do modelo agregado no cenário de aprendizado federado. Na agregação ponderada, os pesos são aplicados estrategicamente sobre os modelos locais de cada dispositivo, levando em conta seu desempenho e sua contribuição efetiva para o aprendizado global. Sabendo que os dispositivos 2 e 4 têm apenas 0,1% de seus dados disponíveis para treinamento, enquanto o dispositivo 1 usa 25% e o dispositivo 3 utiliza 75%, fica claro que atribuir pesos maiores aos dispositivos 1 e 3 é uma decisão coerente com sua maior capacidade de aprendizado. Isso é evidenciado, por exemplo, na configuração

[0.3, 0.2, 0.2, 0.3], que favorece dispositivos 1 e 4. Apesar de 4 ter poucos dados, a presença de pesos médios para o dispositivo 3 garante um F1-Score de 0.851 (CTU) e 0.919 (ISOT), com AUCs também elevados (CTU: 0.769; ISOT: 0.815).

O melhor equilíbrio aparece com pesos que valorizam o dispositivo 3 (com maior dado de treino) e evitam sobrecarregar o dispositivo 2, como em [0.1, 0.4, 0.4, 0.1], que oferece um desempenho elevado (CTU F1: 0.8627; ISOT F1: 0.9207). Já a configuração [0.1, 0.7, 0.1, 0.1], que atribui peso exagerado ao dispositivo 2 (com quase nenhum dado treinado), causa forte queda de desempenho (CTU F1: 0.457; ISOT F1: 0.826), evidenciando o impacto da supervalorização de modelos locais pouco treinados. A agregação ponderada, portanto, se mostra eficaz ao permitir modular a influência de cada dispositivo de acordo com sua qualidade, protegendo o modelo global contra contribuições fracas e favorecendo a robustez.

Já na agregação aritmética, mostrada na tabela 6.4, que ignora intencionalmente as diferenças presentes na distribuição de dados de treinamento. Como visto, os dispositivos 2 e 4 possuem apenas 0,1% dos dados disponíveis para treino, enquanto os dispositivos 1 e 3 têm 25% e 75%, respectivamente. Isso significa que os modelos dos dispositivos 2 e 4 têm desempenho local muito inferior, como demonstram seus F1-scores base. Quando essas contribuições fracas são incluídas sem ajuste de peso, como ocorre na média aritmética, o resultado agregado sofre impacto negativo, mesmo quando há dispositivos fortes na federação.

Capítulo 7

Conclusões

Este trabalho investigou o impacto do aprendizado federado sobre o desempenho do classificador EFC (Energy Flow Classifier), utilizando diferentes estratégias de agregação ponderada entre dispositivos. A avaliação foi conduzida com base nas métricas F1-score e AUC-ROC, com foco na separação entre classes de tráfego de rede benigno e malicioso. A estrutura experimental simulou um ambiente federado com múltiplos dispositivos, cada um treinando o modelo EFC localmente com diferentes volumes de dados, e os resultados foram combinados por meio de agregações com pesos variados.

A Tabela 6.1 sintetizou os efeitos da variação de pesos entre os dois dispositivos, indicando que o desempenho do EFC federado é fortemente sensível à ponderação adotada. As configurações que favoreceram fortemente o dispositivo de melhor desempenho ([0.8,0.2]) produziram classificadores robustos, com separações bem definidas e baixa taxa de erro. Por outro lado, configurações que elevaram o peso do dispositivo menos treinado e com menor F1-score levaram a quedas progressivas no desempenho, mesmo que a AUC-ROC se mantivesse relativamente estável. Isso reforça que, embora a capacidade de ranqueamento do modelo possa resistir a ruídos, a acurácia da fronteira de decisão (capturada pelo F1-score) se deteriora rapidamente quando modelos fracos dominam a agregação.

Nos experimentos com quatro dispositivos, realizados nas bases CTU e ISOT, padrões semelhantes foram observados. No conjunto CTU, os F1-scores individuais variaram de 0,25 a 0,93, enquanto no ISOT variaram de 0,82 a 0,98. Configurações como [0.3, 0.2, 0.2, 0.3] e [0.4, 0.2, 0.1, 0.3] produziram F1-scores elevados (0.8513 e 0.9193) e AUC-ROC superiores a 0.76, comprovando que agregações levemente assimétricas favorecem o desempenho global do EFC. Já combinações como [0.1, 0.7, 0.1, 0.1], que atribuíram grande peso a modelos de baixa precisão, resultaram em quedas expressivas no F1-score (0.4570), demonstrando a fragilidade de agregações mal calibradas.

A divisão dos dados entre os dispositivos simulou condições realistas de aprendizado federado. O dispositivo 1 recebeu 25% dos dados para treino, o dispositivo 3 recebeu 75%, e os dispositivos 2 e 4 receberam apenas 0.1% cada. Esses desequilíbrios foram deliberadamente introduzidos para refletir a variação natural de disponibilidade de dados entre nós em redes federadas. Os resultados mostraram que dispositivos com baixa amostragem comprometem a qualidade do modelo federado quando recebem peso excessivo, mesmo que façam parte da rede. Assim, a estratégia de ponderação deve levar em conta não apenas o F1-score, mas também a representatividade estatística dos dados locais e sua confiabilidade.

Em conclusão, este estudo demonstrou que o modelo federado aplicado ao classificador EFC pode alcançar alto desempenho quando a agregação ponderada é cuidadosamente calibrada. Dar prioridade aos modelos mais robustos, sem excluir completamente os demais, favorece a generalização e a estabilidade do sistema. O equilíbrio ideal se dá entre dominância técnica e diversidade informacional. Para trabalhos futuros, sugere-se investigar estratégias de ponderação adaptativas com base em acurácia validada, volume de treino, incerteza preditiva e impacto marginal. Outro ponto a ser considerado seria a aplicação de reforço, adicionando épocas no aprendizado federado. Além disso, vale explorar o uso do EFC federado em classificações multiclasse e em redes suscetíveis a ataques adversariais, como os descritos por Apruzzese e Colajanni [15] e Zhu et al. [4], ampliando assim a resiliência e aplicabilidade do modelo.

Referências

[1] García, S., M. Grill, J. Stiborek e A. Zunino: *An empirical comparison of botnet detection methods*. Computers Security, 45:100–123, 2014, ISSN 0167-4048. `https://www.sciencedirect.com/science/article/pii/S0167404814000923`. 1, 11

[2] Dao, Nhu Ngoc, Trung V. Phan, Umar Sa'ad, Joongheon Kim, Thomas Bauschert, Dinh Thuan Do e Sungrae Cho: *Securing heterogeneous iot with intelligent ddos attack behavior learning*. IEEE Internet of Things Journal, 16(2):1974 – 1983, 2022. 1, 8, 9, 10, 16, 27

[3] Jielun, Zhang, Liang Shicong, Ye Feng, Qingyang Hu Rose e Qian Yi: *Towards detection of zero-day botnet attack in iot networks using federated learning*. ICC 2023 - IEEE International Conference on Communications, 2023. 1, 2, 7, 8, 15, 27

[4] Zhu, Yuedi, Chao Li e Yong Wang: *Network intrusion detection scheme based on federated learning in heterogeneous network environments*. Em *2024 13th International Conference on Communications, Circuits and Systems (ICCCAS)*, páginas 491–496, 2024. 1, 2, 8, 9, 10, 25, 27, 64

[5] Segun I. Popoola, Ruth Ande, Bamidele Adebisi Guan Gui Mohammad Hammoudeh e Olamide Jogunola: *Federated deep learning for zero-day botnet attack detection in iot-edge devices*. IEEE Internet of Things Journal, 9(05):3930 – 3944, 2022. 1, 2, 7, 8, 9, 10, 14

[6] Li, Jianhua, Lingjuan Lyu, Ximeng Liu, Xuyun Zhang e Xixiang Lyu: *Fleam: A federated learning empowered architecture to mitigate ddos in industrial iot*. IEEE Transactions on Industrial Informatics, 18(6):4059–4068, 2022. 1, 8, 9, 10, 23, 27

[7] Lou, Xingyu, Panda Li, Ning Sun e Guangjie Han: *Botnet intrusion detection method based on federated reinforcement learning*. Em *2023 International Conference on Intelligent Communication and Networking (ICN)*, páginas 180–184, 2023. 1, 2, 8, 10, 20

[8] McMahan, Brendan, Eider Moore, Daniel Ramage, Seth Hampson e Blaise Aguera y Arcas: *Communication-Efficient Learning of Deep Networks from Decentralized Data*. Em Singh, Aarti e Jerry Zhu (editores): *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 de *Proceedings of Machine Learning Research*, páginas 1273–1282. PMLR, 20–22 Apr 2017. `https://proceedings.mlr.press/v54/mcmahan17a.html`. 1, 2, 7, 8, 9, 27

[9] Pontes, Camila, Manuel Souza, Matt Gondim, João Bishop e Marcelo Marotta: *A new method for flow-based network intrusion detection using the inverse potts model*. IEEE Internet of Things Journal, 18(2):1125 – 1136, 2020. 2, 3, 18, 25, 28

[10] Lopes, Daniele A. G., Marcelo A. Marotta, Marcelo Ladeira e João J. C. Gondim: *Botnet detection based on network flow analysis using inverse statistics*. Em *2022 17th Iberian Conference on Information Systems and Technologies (CISTI)*, páginas 1–6, 2022. 2, 11, 19

[11] Popoola, Segun I., Bamidele Adebisi, Mohammad Hammoudeh, Guan Gui e Haris Gacanin: *Hybrid deep learning for botnet attack detection in the internet-of-things networks*. IEEE Internet of Things Journal, 8(6):4944–4956, 2021. 9, 10, 21, 27

[12] Draper-Gil, Gerard, Arash Habibi Lashkari, Mohammad Saiful Islam Mamun e Ali A. Ghorbani: *Characterization of encrypted and vpn traffic using time-related features*. Em *Proceedings of the 2nd International Conference on Information Systems Security and Privacy - ICISSP*, páginas 407–414. INSTICC, SciTePress, 2016, ISBN 978-989-758-167-0. 11

[13] Sharafaldin, Iman, Arash Habibi Lashkari e Ali A. Ghorbani: *Toward generating a new intrusion detection dataset and intrusion traffic characterization*. Em *Proceedings of the 4th International Conference on Information Systems Security and Privacy - ICISSP*, páginas 108–116. INSTICC, SciTePress, 2018, ISBN 978-989-758-282-0. 12

[14] Alenazi, Ahmad, Issa Traore, Karamoko Ganame e Isaac Woungang: *Holistic model for http botnet detection based on dns traffic analysis*. Em *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 10618 de *Lecture Notes in Computer Science*, páginas 1–18. Springer, 2017. 12

[15] Apruzzese, Giovanni e Michele Colajanni: *Evading botnet detectors based on flows and random forest with adversarial samples*. 2018 IEEE 17th International Symposium on Network Computing and Applications (NCA), 2018. 17, 27, 64

[16] Kalakoti, Rajesh, Sven Nõmm e Hayretdin Bahsi: *In-depth feature selection for the statistical machine learning-based botnet detection in iot networks*. IEEE Access, 10:94518–94535, 2022. 24, 28