# ATHENA-FL: Evitando a Heterogeneidade Estatística através do Um-contra-Todos no Aprendizado Federado

**Lucas Airam C. de Souza**[1], **Gustavo F. Camilo**[1], **Gabriel A. Fontes Rebello**[1,2], **Matteo Sammarco**[3], **Miguel Elias M. Campista**[1], **Luís Henrique M. K. Costa**[1]

[1]Grupo de Teleinformática e Automação (GTA)
Universidade Federal do Rio de Janeiro (UFRJ)
[2]Sorbonne Université, CNRS, LIP6, F-75005 Paris, França
[3] Stellantis

***Resumo.*** *O aprendizado federado é um novo paradigma que permite o treinamento de modelos de aprendizado de máquina através da colaboração entre clientes e um servidor de agregação. O treinamento dispensa o compartilhamento de dados privados, garantindo aos clientes privacidade de suas amostras. Entretanto, quando os clientes possuem distribuições de dados distintas, o treinamento apresenta dificuldades de convergência, resultando em erros preditivos no modelo final. Este artigo propõe um sistema de aprendizado federado que considera clientes com distribuições de dados heterogêneas e, mesmo assim, produz modelos acurados em menos épocas de treinamento do que o estado da arte. Os efeitos da heterogeneidade dos dados são mitigados através do agrupamento dos clientes baseado em uma estimativa da distribuição de dados através dos pesos da rede neural treinada localmente. Além disso, o sistema utiliza a técnica um-contra-todos, treina um detector para cada classe no sistema. Assim, grupos diferentes podem combinar os detectores a fim de formar um modelo capaz de detectar classes provenientes de outros grupos. Os resultados mostram que o modelo um-contra-todos possui alta capacidade de identificar corretamente as amostras e com acurácia até 18% maior do que o treinamento tradicional, com um baixo custo de comunicação durante o treinamento, reduzindo a quantidade de bytes transmitidos entre 59,6% até 94% em comparação à arquitetura MobileNet.*

## 1. Introdução

O aprendizado de máquina permite a automação de diversas tarefas através da criação de um modelo que identifica padrões em um conjunto de dados para prever ou classificar novos dados. Entretanto, o treinamento do modelo necessita coletar dados, em alguns casos privados, pois revelam informações sensíveis do usuário ou ponto de coleta. Assim, o aprendizado federado surgiu como uma proposta para o treinamento de modelos de aprendizado de máquina que preserva a privacidade do usuário e dos seus dados.

O treinamento no aprendizado federado substitui o compartilhamento de dados pelo compartilhamento de parâmetros do modelo, sendo o FedAVG (*Federated Averaging*) o algoritmo mais utilizado [McMahan et al. 2017]. No FedAVG, os clientes treinam o modelo localmente por algumas épocas e enviam o resultado para um servidor de agregação, que combina as respostas individuais em um modelo global. Então, o servidor de agregação retransmite o modelo global para que os clientes possam aprimorá-lo, sendo este processo repetido até que o modelo global convirja ou o número de épocas globais seja alcançado. O FedAVG é um algoritmo específico para modelos de aprendizado de máquina federado baseado na atualização de

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001, CNPq, FAPERJ (E-26/203.211/2017, E-26/202.932/2017, E-26/202.689/2018 e E-26/200.892/2021) e FAPESP (15/24494-8, 18/23292-0, 15/24485-9 e 14/50937-1.)

parâmetros através do vetor gradiente de perda, e calcula a média dos vetores enviados pelos clientes para atualizar o modelo global. Assim, as amostras dos clientes permanecem armazenadas localmente, preservando a privacidade dos dados e do usuário. Entretanto, quando os clientes possuem dados heterogêneos, distribuições não independentes e identicamente distribuídas (não-IID), o modelo possui dificuldade de convergência e baixo desempenho final.

O agrupamento de clientes pela similaridade dos dados permite que os modelos sejam treinados com dados IID e convirjam rapidamente com alto desempenho final de classificação [Ouyang et al. 2021]. Porém, as arquiteturas de aprendizado profundo treinadas para o problema de classificar múltiplas classes nos grupos não são generalizáveis para dados provenientes de outras distribuições. Portanto, torna-se necessária uma proposta que reduza os efeitos da heterogeneidade dos dados durante o treinamento dos modelos e permita a generalização do classificador para amostras originadas por outras distribuições de dados.

Este artigo propõe o ATHENA-FL[1], um sistema que utiliza o treinamento de modelos com a técnica um-contra-todos (*One-versus-All* - OvA)[2] para permitir o compartilhamento de modelos entre grupos. Além disso, o sistema reduz a heterogeneidade estatística através do agrupamento de clientes pela similaridade dos dados. O modelo um-contra-todos utiliza classificadores binários, treinados de forma independente, que estimam a probabilidade da amostra pertencer à classe identificada pelo detector. Após o treinamento, os detectores são combinados para a classificação de amostras. Cada detector estima a probabilidade da amostra pertencer à sua classe e o classificador rotula a amostra a partir do detector que gera a maior probabilidade. Assim, o método um-contra-todos é utilizado para o compartilhamento eficiente de modelos entre grupos a fim de criar um modelo genérico para a classificação de dados provenientes de diferentes grupos. Além disso, o sistema agrupa os clientes de acordo com suas distribuições de dados para que os detectores convirjam rapidamente com um alto desempenho de classificação.

Os experimentos realizados avaliam a evolução da acurácia dos detectores e comparam o desempenho de classificação e os requisitos de comunicação para o treinamento dos modelos. A acurácia dos modelos depende da diversidade das amostras utilizadas para o treinamento dos detectores, com a acurácia do modelo um-contra-todos até 18% maior do que a arquitetura MobileNet. Simultaneamente, a quantidade de bytes transmitidos ao longo de todas as épocas de treinamento é reduzida em até 94% utilizando o modelo um-contra-todos ao invés da MobileNet. Assim, o sistema oferece uma forma eficaz de treinar modelos que aumentam a privacidade dos dados dos clientes, mesmo em cenários com distribuições de dados heterogêneas.

Este artigo está organizado da seguinte forma. A Seção 2 revisa o estado da arte em propostas para aumentar o desempenho de classificação de sistemas de aprendizado federado e reduzir os impactos causados pela heterogeneidade dos dados. A Seção 3 descreve o sistema ATHENA-FL proposto. A Seção 4 apresenta o desenvolvimento de um protótipo do ATHENA-FL e a análise dos resultados obtidos. Por fim, a Seção 5 conclui este trabalho e discute direções de pesquisa futuras.

## 2. Trabalhos Relacionados

Atualmente, um dos desafios mais relevantes em relação ao aprendizado federado é aumentar o desempenho final do modelo e reduzir o tempo total de treinamento. Apesar da otimização de hiperparâmetros ser uma alternativa promissora [Neto et al. 2021], a seleção de clientes [Fu et al. 2022] ou a personalização dos modelos locais [Tan et al. 2022] são alternativas mais adotadas para obter a convergência mais rapidamente. Enquanto a proposta inicial de

[1]Acrônimo do inglês: Avoiding sTatistical HEterogeinety with oNe-versus-All in Federated Learning.

[2]Disponível em https://github.com/GTA-UFRJ/ATHENA-FL.

aprendizado federado considera uma distribuição uniforme para a probabilidade de um cliente participar do treinamento [McMahan et al. 2017], a seleção de clientes modifica essa função de probabilidade para refletir a capacidade do cliente contribuir significativamente para o treinamento do modelo. Por outro lado, a personalização de modelos permite que os modelos sejam individuais e ajustados localmente após o treinamento global ter atingido um limiar de desempenho predefinido. Dessa forma, a seleção de clientes aumenta o desempenho global do modelo, enquanto a personalização aumenta o desempenho local.

### 2.1. Seleção de Clientes para o Treinamento Eficiente

Luo *et al.* e Lai *et al.* propõem esquemas de seleção de clientes que buscam otimizar a velocidade de convergência do modelo em ambientes de aprendizado federado [Luo et al. 2022, Lai et al. 2021]. A seleção baseada apenas na representatividade dos dados diminui o número total de épocas para convergência do modelo. Entretanto, clientes que possuem dados mais relevantes para o problema podem apresentar um tempo maior de treinamento para cada época, tanto em relação à quantidade de dados quanto em relação ao tipo de *hardware* utilizado. Aumentar o tempo entre épocas implica um atraso geral maior, pois ambientes de aprendizado federado geralmente são síncronos e aguardam a resposta de todos os clientes ou o tempo limite de espera. Por outro lado, selecionar clientes com maior capacidade computacional para reduzir o tempo entre épocas pode incorrer em um número maior de épocas para a convergência, caso os clientes selecionados possuam dados pouco significativos estatisticamente para a tarefa de aprendizado. Portanto, existe um compromisso entre a quantidade de épocas para o treinamento dos modelos e o tempo total de cada época. Assim, os autores consideram simultaneamente as características dos dispositivos e as distribuições dos dados coletados, para reduzir o tempo de convergência do modelo global.

Wang *et al.* utilizam o aprendizado por reforço para seleção eficiente dos clientes [Wang et al. 2020a]. O agente do aprendizado por reforço é ajustado para selecionar os melhores clientes a cada época de treinamento e prover o melhor grupo de clientes dado o estado atual do modelo global. As informações estatísticas dos clientes nas propostas apresentadas são estimadas através do vetor de gradiente de perda enviado ao servidor de agregação para manter a privacidade dos dados dos clientes. Fu *et al.* discutem os principais sistemas e arcabouços de seleção de clientes no aprendizado federado [Fu et al. 2022]. O artigo apresenta o estado da arte na seleção de clientes e compara suas principais diferenças, demonstrando que a seleção de clientes possui grande potencial para produzir modelos mais acurados no aprendizado federado com menor tempo de treinamento.

### 2.2. Personalização de Modelos no Aprendizado Federado

FedTP (*Federated learning by Transformer Personalization*) é um arcabouço para a redução da heterogeneidade dos dados por transformações dos conjuntos de dados e personalização dos modelos [Li et al. 2022]. O objetivo da proposta é definir a base de uma transformação para que os dados dos clientes sejam similares e que os modelos possam ser personalizados em algumas camadas e, assim, contornar os problemas de convergência. Entretanto, a proposta gera sobrecarga na comunicação, exigindo um alto tempo para o ajuste dos parâmetros da projeção e para a convergência do modelo.

O trabalho [de Souza et al. 2022] propõe um sistema de aprendizado federado através do agrupamento de clientes. Após o agrupamento dos clientes, cada grupo treina seu próprio modelo de forma federada. Entretanto, a proposta apenas inclui modelos específicos e o conhecimento é mantido nos grupos. Este trabalho estende a proposta para prover uma forma de combinar os modelos gerados em diferentes grupos através da criação de um modelo OvA.

Zhu *et al.* propõem o uso do esquema de classificação um-contra-todos para mitigar o impacto dos dados heterogêneos no treinamento de modelos de aprendizado federado [Zhu et al. 2021]. O algoritmo FedOVA (*Federated OvA*) utiliza modelos para classificação binária de amostras e seleciona os clientes que possuem amostras da classe alvo para realizar o treinamento. Entretanto, a proposta não estuda o impacto da criação dos modelos e não possui um protocolo adequado para o treinamento dos detectores. O ATHENA-FL agrupa os clientes conforme as distribuições dos dados, antes de realizar o treinamento do modelo OvA e, dessa forma, reduz o tempo de treinamento dos detectores.

FLEE (*Federated Learning Early Exit of inference*) é um arcabouço de aprendizado federado hierárquico que divide o modelo em três localizações diferentes [Zhong et al. 2022]. A divisão do modelo entre a nuvem, borda e dispositivo final permite utilizar o método de saídas antecipadas de rede neural no processo de inferência. Além disso, a divisão hierárquica do treinamento reduz o impacto causado por distribuições de dados não-IID na convergência do modelo, pois os autores assumem que os dados possuem maior similaridade segundo a distância geográfica dos clientes.

CEFL (*Communication-Efficient Federated Learning*) é um arcabouço para o treinamento de modelos para dados médicos através de aprendizado federado [Chu et al. 2022]. Os autores propõem determinar a similaridade entre os clientes através do cálculo da distância euclidiana dos pesos das redes neurais dos clientes. Baseado na similaridade, os clientes são agrupados utilizando o método de Louvain [Blondel et al. 2008]. Em cada grupo, o cliente com maior soma de similaridade é escolhido para ser o líder. O papel do líder é realizar o treinamento federado das primeiras camadas do modelo em conjunto com líderes de outros grupos, enquanto as camadas finais do modelo são treinadas de forma personalizada em cada grupo.

Diferente das propostas anteriores, este artigo propõe o ATHENA-FL, um sistema de aprendizado federado baseado no agrupamento de clientes por similaridade e no treinamento de modelos de redes neurais através da abordagem um-contra-todos. Inicialmente os clientes são agrupados utilizando o peso das redes neurais de cada cliente como um vetor de entrada para o algoritmo de agrupamento. A similaridade aumenta a acurácia e reduz o tempo de treinamento dos modelos de cada grupo, pois torna os dados de treinamento mais homogêneos. Cada grupo treina detectores das classes presentes nos conjuntos de dados dos clientes. Ao fim do treinamento dos modelos de cada grupo, é possível combiná-los para detectar a classe de amostras que não pertencem ao grupo através da utilização do modelo um-contra-todos. O desempenho do ATHENA-FL é comparado com o aprendizado federado tradicional utilizando a arquitetura MobileNet, uma rede neural profunda leve para classificação de imagens. Além disso, os experimentos comparam o custo de comunicação das duas abordagens, calculado através da quantidade de bytes transmitidos por cliente durante o treinamento. Entretanto, há o compromisso entre a quantidade de classes existentes e a acurácia do modelo um-contra-todos. Mais classes no sistema, implica mais detectores, que devem ser treinados com dados mais diversos para uma melhor distinção entre as classes. Assim, detectores treinados com poucas classes apresentam um desempenho menor em dados fora do grupo, reduzindo a acurácia do modelo OvA.

## 3. O Sistema Proposto: ATHENA-FL

O ATHENA-FL é um sistema para treinamento de modelos sob o paradigma de aprendizado federado. No sistema proposto, os clientes são agrupados conforme a similaridade dos dados, e utiliza o modelo um-contra-todos para problemas de classificação de amostras. O agrupamento de clientes permite a criação de modelos de aprendizado de máquina de forma mais

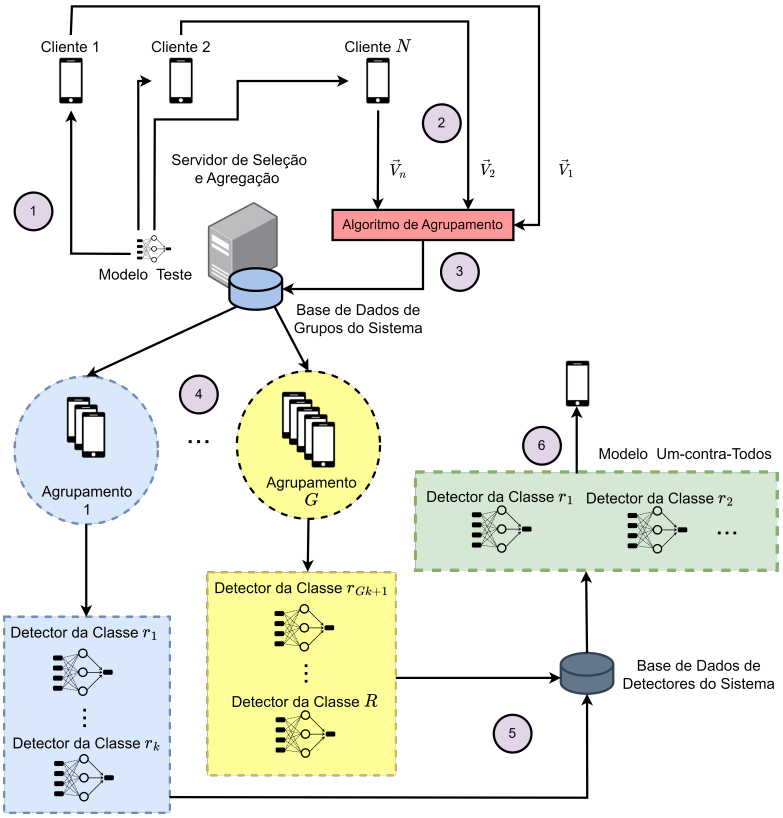



**Figura 1: Esquema de execução do ATHENA-FL composto por 6 etapas de operação. As Etapas de 1 a 3 têm o objetivo de mitigar a heterogeneidade estatística do sistema agrupando os clientes segundo a similaridade dos dados. Na Etapa 4, os clientes treinam os modelos nos grupos, enquanto a Etapa 5 descreve o compartilhamento dos modelos detectores das classes existentes em cada grupo. Por fim, na Etapa 6, um cliente pode combinar os modelos de diferentes grupos para criar um modelo robusto que utiliza o método um-contra-todos.**

eficiente do que a proposta inicial de aprendizado federado [McMahan et al. 2017], aumentando a acurácia final e reduzindo o total de épocas para a convergência do modelo, pois os grupos contêm clientes com dados similares. Por outro lado, o agrupamento cria modelos específicos para classificar as amostras do seu grupo de treinamento. Para incluir a possibilidade de criação de modelos genéricos através do compartilhamento de modelos entre os grupos, o sistema proposto adota o método OvA. O modelo de aprendizado um-contra-todos adota classificadores específicos para cada tipo de amostra, capazes de serem treinados em poucas épocas, que identificam se um dado pertence a uma classe. A combinação dos modelos detectores permite a criação de um classificador genérico, capaz de identificar diferentes classes. A Figura 1 exibe as etapas propostas para a execução do sistema.

Assim, esta seção apresenta os detalhes da proposta para o agrupamento de clientes e o treinamento dos modelos. Primeiramente são discutidas as hipóteses iniciais sobre a distribuição de dados e condições do sistema. Em seguida, a seção aborda os detalhes do agrupamento de clientes para a criação de modelos específicos. Então, exibem-se as etapas realizadas para treinar os modelos em cada grupo. Por fim, discute-se como formar modelos mais genéricos a partir da combinação dos modelos específicos gerados nos grupos.

### 3.1. Hipóteses do Ambiente de Execução

Assume-se que os grupos são formados de forma homogênea, sendo assim, os clientes possuem amostras similares das mesmas classes. O grupo treina detectores para todas as classes existentes nos conjuntos de dados dos clientes. Além disso, o servidor em conjunto com os clientes identifica de forma única as tarefas de classificação e os rótulos existentes. Dessa forma, novas classes podem ser incorporadas ao sistema sem perda de generalidade em relação aos modelos existentes. Além disso, não há rótulos iguais para amostras pertencentes a classes diferentes no sistema antes do agrupamento.

### 3.2. Agrupamento dos Clientes Através da Similaridade dos Dados

Inicialmente, o sistema agrupa os clientes conforme a similaridade dos dados, para que o treinamento dos modelos seja sobre dados que atendam a suposição de que são independentes e identicamente distribuídos. Conjuntos de dados IID facilitam a convergência do modelo e aumentam o desempenho de classificação final [Wang et al. 2020b]. Entretanto, é inviável obter diretamente dos clientes qual a sua distribuição de dados para o agrupamento devido às restrições impostas sobre a privacidade dos dados. Assim, o sistema utiliza os pesos de um modelo teste, treinado com os dados sensíveis, para identificar os diferentes tipos de dados existentes.

O sistema executa as Etapas 1 a 4, exibidas na Figura 1, para agrupar os clientes conforme a similaridade dos dados. Primeiramente, um modelo genérico de teste é distribuído para todos os clientes do sistema. Dessa forma, os clientes ajustam os pesos da rede neural, que dependem dos dados privados. Assim, após realizar o treinamento local, cada cliente possui a mesma estrutura de rede neural, porém com pesos diferentes. Os pesos da rede neural são compartilhados com o servidor de agregação na Etapa 2 do diagrama. Como os pesos possuem correlação com os dados privados, usuários com dados similares possuem pesos próximos e são consequentemente indicados para o mesmo grupo na Etapa 3. Para identificar os grupos, o sistema utiliza o algoritmo de agrupamento espacial de aplicações com ruído baseado na densidade (*Density-Based Spatial Clustering of Applications with Noise* - DBSCAN) [Ester et al. 1996], como realizado anteriormente [de Souza et al. 2022]. Diferente dos detectores, o modelo utilizado para teste é um modelo profundo para classificação de múltiplas classes, pois é necessário comparar as distribuições de dados entre todos os clientes nessa etapa. O processo de agrupamento se encerra com o início da Etapa 4, que realiza o treinamento dos detectores. Novos clientes, ausentes na etapa de criação dos grupos, devem solicitar ao servidor a sua alocação em um grupo, através da execução das Etapas 3 e 4 com o modelo de agrupamento configurado com todos os grupos existentes.

### 3.3. Treinamento dos Modelos Detectores

Os modelos detectores são treinados de forma independente, tanto em relação aos modelos de grupos diferentes, quanto em relação aos detectores treinados dentro de cada grupo. Assim, esta etapa pode ser paralelizada para reduzir o tempo de treinamento dos modelos. Cada grupo possui um total de $k$ detectores, que pode variar segundo o grupo, e o sistema possui $R$ detectores no total.

Assim, na Etapa 4 da Figura 1 ocorre o treinamento dos detectores de cada grupo presente no sistema. O treinamento se inicia com o servidor de seleção e agregação determinando o detector que será treinado de forma federada em um grupo. Após a definição da classe de interesse, o servidor envia esta informação para que os clientes realizem o pré-processamento dos rótulos. O pré-processamento consiste em transformar os rótulos originais de todas as

amostras dos clientes em rótulos binários, onde todas as classes recebem o rótulo zero, exceto a classe de interesse. Por exemplo, para um problema de classificação com quatro classes $C = \{r_1, r_2, r_3, r_4\}$, cuja classe de interesse possua o rótulo $r_2$, o pré-processamento mapeia os rótulos originais de todas as amostras para o conjunto $C_{bin_2} = \{0, 1, 0, 0\}$. A partir desta etapa o detector da classe $r_2$ é treinado através do aprendizado federado no grupo.

O tempo de convergência para cada detector é reduzido devido ao agrupamento de clientes que possuem conjuntos de dados aproximadamente IID. Porém, os modelos gerados no grupo são específicos e detectam apenas as classes existentes no grupo. Assim, após a convergência do modelo de um detector, ele é disponibilizado na base de dados de detectores do sistema. Esta é a Etapa 5 da Figura 1. Isso permite que clientes de outros grupos possam construir modelos um-contra-todos com detectores gerados em outros grupos.

Pode-se destacar duas vantagens em utilizar a abordagem: os modelos utilizados pelos detectores são menores, logo o tempo entre diferentes épocas de treinamento é reduzido em relação às redes neurais mais profundas, e a criação de um modelo robusto, capaz de identificar as classes existentes, é realizada de forma simples, pois basta combinar os modelos criados em diferentes grupos.

### 3.4. Criação de Modelos Genéricos

A etapa final da operação do sistema, Etapa 6, permite a criação de modelos genéricos, capazes de utilizar detectores gerados em outros grupos. Após o compartilhamento dos modelos na base de dados de detectores do sistema, os clientes podem selecionar os detectores de seu interesse. Os detectores selecionados são combinados para formar o modelo $C_{ova}$ do cliente. A Equação 1 exibe o processo de classificação de uma amostra $x$ pelo modelo $C_{ova}$.

$$C_{ova}(x) = \underset{i \in [1,R]}{\operatorname{argmax}}\, r_i(x). \tag{1}$$

O modelo é composto por diversos detectores $r_i$, que classificam a amostra $x$ e retornam a probabilidade da amostra pertencer à classe $i$. O classificador $C_{ova}$ atribui à amostra o rótulo que possui a maior probabilidade entre todos os detectores avaliados. O processo de classificação é paralelizável, pois cada modelo pode simultaneamente gerar a probabilidade da amostra pertencer a uma classe, uma vez que os modelos são independentes.

## 4. Desenvolvimento do Protótipo e Resultados Obtidos

Um protótipo do ATHENA-FL foi desenvolvido utilizando a linguagem de programação Python v3.9.1 com o arcabouço flower v1.1.0 para a construção do ambiente de aprendizado federado e a biblioteca scikit-learn v1.0.2 para a criação dos modelos de agrupamento. Os experimentos deste trabalho foram realizados em um servidor Intel Xeon CPU E5-2650 2.00 GHz com 32 núcleos de processamento e 504 GB de RAM. Os resultados experimentais da avaliação dos modelos apresentam a média obtida entre todos os clientes. Como os gráficos apresentam múltiplos resultados, o intervalo de confiança de 95% foi omitido das figuras para facilitar a visualização.

Os cenários avaliados utilizam os valores de configuração exibidos na Tabela 1. A probabilidade de seleção exibida na tabela equivale ao percentual de clientes selecionados dentro de cada grupo para o treinamento. O número de clientes varia nos cenários avaliados. Além disso, os resultados de acurácia são obtidos a partir da seleção de todos os clientes a cada época global. O algoritmo de agrupamento é definido com o parâmetro de distância

**Tabela 1: Principais parâmetros aplicados ao ambiente de aprendizado federado.**

| Parâmetro | Configuração |
|---|---|
| Probabilidade de Seleção de Cliente | 20% |
| Número de Épocas Globais | 200 |
| Número de Épocas Locais | 5 |
| Tamanho dos Lotes Locais | 15 amostras |

$d = 0.0279$ e número mínimo de clientes por grupo igual a 2, como analisado em resultados anteriores [de Souza et al. 2022].

A arquitetura de rede neural utilizada no modelo um-contra-todos foi adaptada de um problema de identificação de imagens de cães e gatos, estabelecendo um modelo simples para classificação binária. Então, essa arquitetura de rede neural foi adaptada para ser a base dos detectores utilizados no modelo um-contra-todos devido ao seu tamanho pequeno e à sua alta capacidade de identificar amostras corretamente.

As arquiteturas profundas, utilizadas em tarefas de classificação com múltiplas classes, são: MobileNet, MobileNetV2 e a Xception. A MobileNetV2 foi escolhida por ser a arquitetura utilizada no exemplo de classificação do conjunto de dados CIFAR-10 no arcabouço flower e a MobileNet por possuir um tempo de inferência menor do que a MobileNetV2 por possuir uma arquitetura menos profunda. Por fim, a Xception foi escolhida pelo seu alto desempenho de classificação em conjuntos de dados de imagem.

O treinamento e avaliação de desempenho dos modelos foram realizados sobre dois conjuntos de dados de imagens: CIFAR-10 [Krizhevsky et al. 2009] e o MNIST [LeCun et al. 2010]. O conjunto de dados CIFAR-10 possui 60.000 amostras e o total de 10 classes que representam objetos ou animais. As imagens são coloridas com 3 canais RGB e possuem 32x32 píxeis. O segundo conjunto de dados avaliado, MNIST, possui 70.000 amostras divididas em 10 classes, que representam dígitos decimais escritos à mão. As imagens são processadas em escala de cinza e possuem originalmente 28x28 píxeis, expandidas em 32x32 píxeis para utilizar a mesma arquitetura de rede neural nos dois conjuntos de dados. Ambos os conjuntos de dados são balanceados em relação às classes existentes. Portanto, o experimento consiste em duas etapas: a avaliação da acurácia dos detectores 4.1, determinando também a quantidade de épocas necessárias para a convergência dos modelos, e a avaliação da comunicação 4.2, que estima a eficiência do sistema proposto em relação à quantidade de bytes enviados em comparação aos modelos profundos.

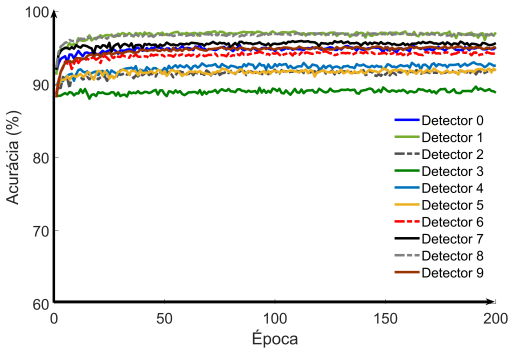
### 4.1. Avaliação da Acurácia

O sistema proposto é avaliado em distribuições de dados IID e não-IID. As distribuições de dados não-IID são consideradas em dois cenários, no primeiro cada cliente possui amostras de apenas duas classes do conjunto de dados, e o segundo considera clientes com amostras de cinco classes diferentes.

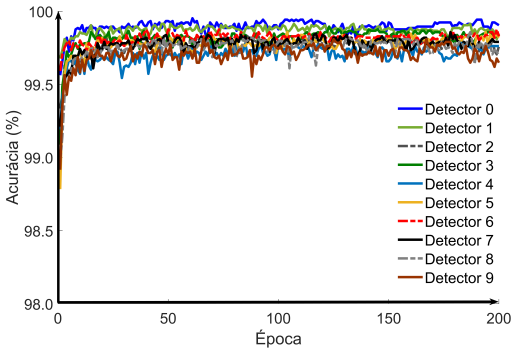
#### 4.1.1. Cenário com Dados IID

No primeiro experimento, os clientes possuem conjuntos de dados IID. Neste caso, os conjuntos de dados utilizados são distribuídos igualmente entre todos os clientes, mantendo o balanceamento original entre as diversas classes. O algoritmo de agrupamento define todos os clientes como pertencentes ao mesmo grupo, pois o conjunto de dados é homogêneo. Este

experimento utiliza 20 clientes para o treinamento federado. A Figura 2 exibe os resultados de acurácia para essa configuração.



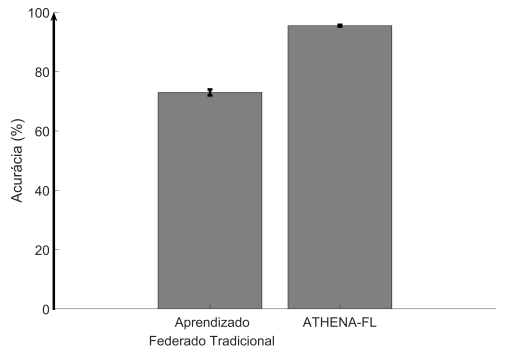

(a) Evolução da acurácia de teste dos detectores ao longo das épocas de treinamento utilizando o conjunto de dados CIFAR-10.



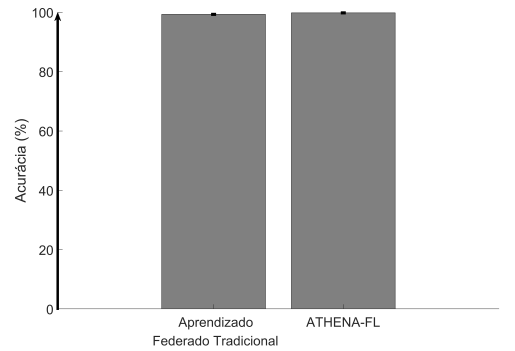

(b) Acurácia dos detectores ao longo das épocas de treinamento utilizando o conjunto de dados MNIST.

**Figura 2: Avaliação da acurácia dos detectores nos conjuntos de dados. Os clientes possuem distribuições de dados IID.**

Os detectores utilizados convergem rapidamente para um alto valor de acurácia e o desempenho final do modelo um-contra-todos para o conjunto de dados CIFAR-10 é de $(95,5 \pm 0,3)\%$, enquanto a arquitetura MobileNet apresenta uma acurácia final de $(73 \pm 1)\%$. No conjunto de dados MNIST as abordagens apresentam um desempenho mais próximo, com $(99.3 \pm 0.1)\%$ e $(99.8 \pm 0.1)\%$ para o aprendizado federado tradicional e o ATHENA-FL, como exibe a Figura 3. Assim, o experimento mostra a vantagem de utilizar o modelo um-contra-todos no aprendizado federado, mesmo em casos nos quais os clientes possuem distribuições de dados homogêneos. A variação de desempenho entre os dois conjuntos de dados é devido à dificuldade do problema apresentado por cada um. O MNIST possui imagens mais simples em escala de cinza, sendo mais simples para a classificação. Logo, os detectores possuem maior acurácia e menor variação nesse conjunto de dados do que no CIFAR-10 que possui imagens coloridas com mais elementos. Observa-se também esse comportamento nos outros cenários avaliados.




(a) Acurácia final dos modelos utilizando o conjunto de dados CIFAR-10.

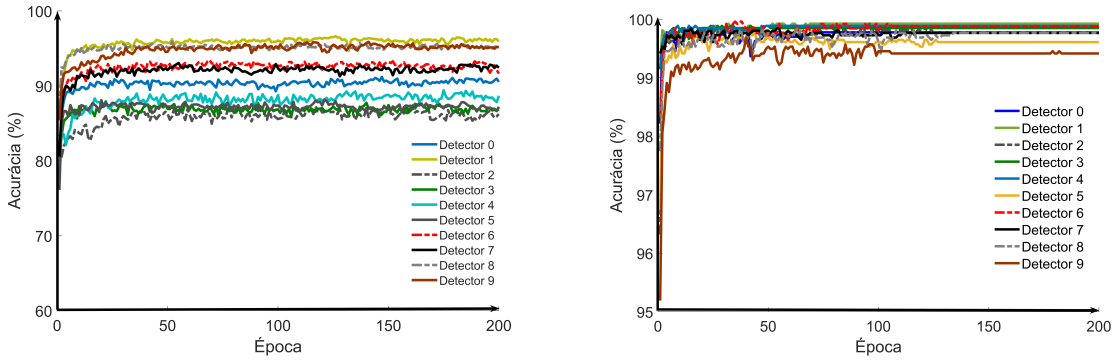



(b) Acurácia final dos modelos utilizando o conjunto de dados MNIST.

**Figura 3: Acurácia final dos modelos no aprendizado federado. Os clientes possuem distribuições de dados IID.**

### 4.1.2. Cenário com Dados não-IID

O cenário com dados não-IID considera distribuições de dados em que os clientes possuem apenas um subconjunto das classes do conjunto de dados. No primeiro caso não-IID, os clientes possuem amostras de cinco classes distintas. Esse experimento considera um cenário com 40 clientes para o treinamento federado, divididos em dois grupos pelo algoritmo de agrupamento.

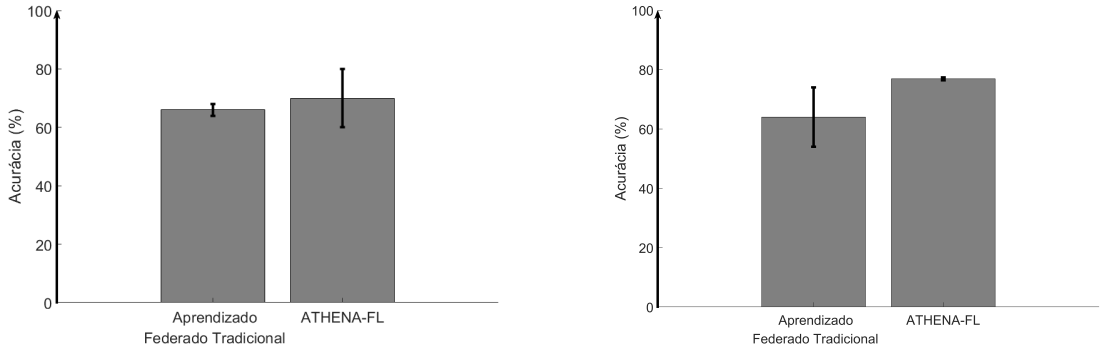



(a) Evolução da acurácia de teste dos detectores ao longo das épocas de treinamento utilizando o conjunto de dados CIFAR-10.

(b) Acurácia dos detectores ao longo das épocas de treinamento utilizando o conjunto de dados MNIST.

**Figura 4: Avaliação da acurácia dos detectores nos conjuntos de dados. Os clientes possuem distribuições de dados não-IID com amostras de 5 classes.**

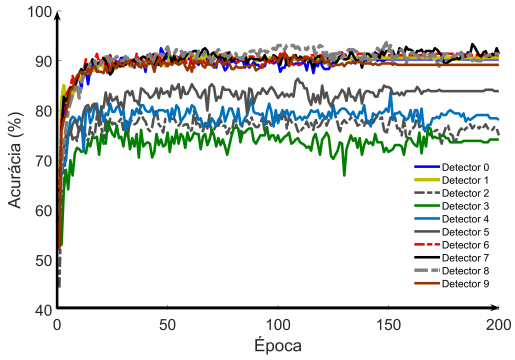
A Figura 4 exibe a variação da acurácia dos detectores em função das épocas de treinamento para o caso não-IID com cinco classes distintas por clientes. O desempenho final do modelo um-contra-todos para o conjunto de dados MNIST é de $(76,9 \pm 0,5)\%$, enquanto a arquitetura MobileNet apresenta uma acurácia final de $(64 \pm 10)\%$ nesta configuração. O desempenho final do modelo um-contra-todos para o conjunto de dados CIFAR-10 é de $(70 \pm 10)\%$, enquanto a arquitetura MobileNet apresenta uma acurácia final de $(66 \pm 2)\%$ na mesma configuração. A Figura 5 exibe o desempenho final de classificação dos sistemas para os dois conjuntos de dados utilizados. Assim, o experimento demonstra a capacidade do modelo um-contra-todos aumentar em média 4% o desempenho de classificação em comparação com o modelo treinado MobileNet no aprendizado federado com dados heterogêneos.



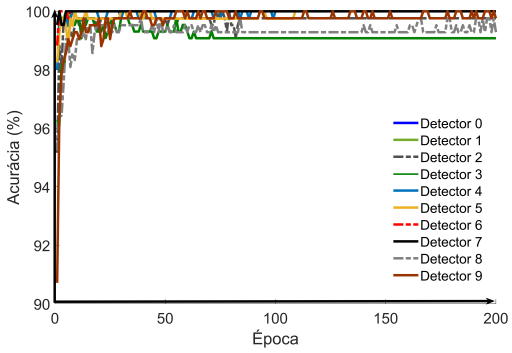

(a) Acurácia final dos modelos utilizando o conjunto de dados CIFAR-10.

(b) Acurácia final dos modelos utilizando o conjunto de dados MNIST.

**Figura 5: Acurácia final dos modelos no aprendizado federado. Os clientes possuem distribuições de dados não-IID com amostras de 5 classes.**



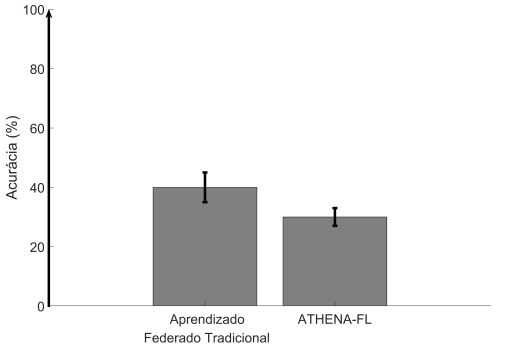

(a) Evolução da acurácia de teste dos detectores ao longo das épocas de treinamento utilizando o conjunto de dados CIFAR-10.



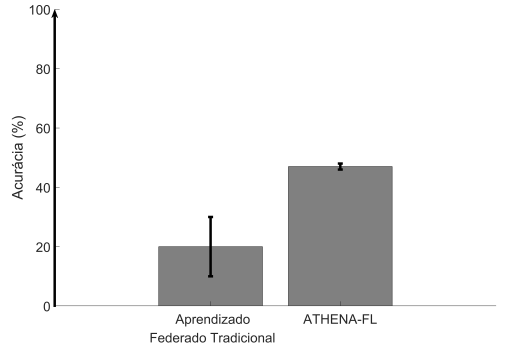

(b) Acurácia dos detectores ao longo das épocas de treinamento utilizando o conjunto de dados MNIST.

**Figura 6: Avaliação da acurácia dos detectores nos conjuntos de dados. Os clientes possuem distribuições de dados não-IID com amostras de duas classes.**

O último cenário considera uma distribuição de dados não-IID com duas classes por cliente. A Figura 6 demonstra o desempenho dos classificadores ao longo das épocas de treinamento. Esse experimento utiliza 20 clientes por grupo para o treinamento federado, com 5 grupos criados pelo algoritmo de agrupamento e um total de 100 clientes no sistema. O desempenho final do modelo um-contra-todos para o conjunto de dados MNIST foi de $(47 \pm 1)\%$, enquanto o modelo profundo obteve uma acurácia de $(20 \pm 10)\%$. Para o conjunto de dados CIFAR-10, a acurácia observada é de $(30 \pm 3)\%$ combinando os detectores, enquanto a arquitetura MobileNet apresenta uma acurácia final de $(40 \pm 5)\%$ nessa configuração. A Figura 7 apresenta os resultados de acurácia final das duas abordagens. O baixo desempenho de classificação nesse cenário deve-se à pouca diversidade de amostras. Os resultados mostram que quando os detectores são treinados em conjuntos de dados com mais classes, o desempenho final de classificação é melhor. Esse comportamento pode ser explicado pela maior variância dos dados das classes que não são de interesse do detector. A maior variância dos dados em outras classes permite o detector identificar características mais relevantes nos dados de interesse, ao invés de apenas diferenciar características específicas da imagem e não representativas para o problema.




(a) Acurácia final dos modelos utilizando o conjunto de dados CIFAR-10.




(b) Acurácia final dos modelos utilizando o conjunto de dados MNIST.

**Figura 7: Acurácia final dos modelos no aprendizado federado. Os clientes possuem distribuições de dados não-IID com amostras de duas classes.**

### 4.2. Avaliação da Comunicação

O objetivo deste experimento é avaliar o custo da comunicação entre os clientes e o servidor de agregação durante o treinamento dos modelos um-contra-todos e da rede neural profunda para classificar múltiplas classes. A avaliação da comunicação considera o total de bytes transmitidos em média para realizar o treinamento do modelo. Sejam $T_{dec}$ o tamanho em bytes do detector, $e_{dec}$ a função de densidade de probabilidade que indica a quantidade de épocas necessárias para a convergência do detector e $R$ a quantidade de detectores existentes no problema de classificação, a quantidade média de bytes transmitidos por cliente $\overline{B_{ova}}$ pode ser expressa pela relação 2 para o modelo de aprendizado um-contra-todos:

$$\overline{B_{ova}} = T_{dec} \times R \times \mathbb{E}[e_{dec}]. \tag{2}$$

Por outro lado, o custo de comunicação das arquiteturas de redes neurais para classificação de múltiplas classes $B_{cmc}$ é dado por $\overline{B_{cmc}} = T_{cmc} \times \mathbb{E}[e_{cmc}]$. $T_{cmc}$ é o tamanho e $e_{cmc}$ é a função de densidade de probabilidade que indica a quantidade de épocas necessárias para a convergência do modelo de classificação de múltiplas classes. Os valores de $T_{dec}$ e $T_{cmc}$ são determinísticos e dependem da arquitetura de rede neural utilizada. A Figura 8 exibe a comparação entre o tamanho em bytes das diferentes arquiteturas de redes neurais avaliadas neste trabalho. Usando a biblioteca Keras para a criação dos modelos, cada detector possui $T_{dec} = 551$k bytes. Por outro lado, a arquitetura MobileNetV2 possui 9,2MB, enquanto a MobileNet e a Xception possuem 13MB e 81MB, respectivamente. A quantidade de detectores também é um valor determinístico que depende apenas do conjunto de dados utilizado para avaliação, que nos casos testados possui o valor $R = 10$. Além disso, os valores esperados de $e_{dec}$ e $e_{cmc}$, $\mathbb{E}[e_{dec}]$ e $\mathbb{E}[e_{cmc}]$, são obtidos experimentalmente, através do desempenho de treinamento ao longo das épocas. Quando o modelo não apresenta uma melhora significativa em relação às épocas anteriores, o treino é considerado finalizado. Analisando os custos apresentados, para que o custo de comunicação do modelo um-contra-todos seja menor ou equivalente ao modelo de rede neural de múltiplas classes, é necessário que as quantidades totais de bytes transmitidos para o treinamento dos modelos sejam relacionadas da seguinte forma: $\overline{B_{ova}} \leq \overline{B_{cmc}}$. Para verificar a validade da relação e assim, a eficiência de comunicação do modelo um-contra-todos, é necessário estimar os valores de $\mathbb{E}[e_{dec}]$ e $\mathbb{E}[e_{cmc}]$.

O critério utilizado para determinar a época a qual o modelo convergiu consiste na comparação da acurácia na época atual em relação à acurácia na época anterior. Caso a época esteja no limite de $tol = 0.1\%$ em relação à época anterior, é considerado que o modelo convergiu. O valor de $tol$ adotado permite que seja considerado apenas casos em que o modelo está estável em um intervalo curto. Aumentar o valor de $tol$ implica reduzir a quantidade de épocas necessárias para a convergência, porém uma queda no desempenho de classificação final. O oposto ocorre quando o seu valor é reduzido. A análise da acurácia ao longo das épocas de treinamento dos resultados, apresentados na Seção 4.1.1 segundo o critério de convergência adotado, permite a estimativa do valor $\mathbb{E}[e_{dec}]$ e $\mathbb{E}[e_{cmc}]$ para os conjuntos de dados em cada uma das configurações experimentais. Para determinar a economia em bytes transmitidos no pior caso entre o modelo um-contra-todos e a arquitetura MobileNet, foi utilizada a relação: $1 - \frac{(\overline{e_{ova}} + \sigma(e_{ova})) \times T_{ova} \times R}{(\overline{e_{cmc}} - \sigma(e_{cmc})) \times T_{cmc}}$. Aplicando o critério de convergência para o caso IID, os detectores convergem em $(7 \pm 5)$ para o conjunto de dados CIFAR-10, enquanto $(4 \pm 2)$ para o conjunto de dados MNIST. Além disso, o modelo profundo MobileNet é treinado com o conjunto de dados CIFAR-10 em média por $(23 \pm 10)$ épocas para convergir. Assim, a utilização do modelo um-contra-todos apresenta uma economia de comunicação de $1 - \frac{(7+5).10.551.2^{10}}{(23-10).9,2.2^{20}} \approx 62\%$

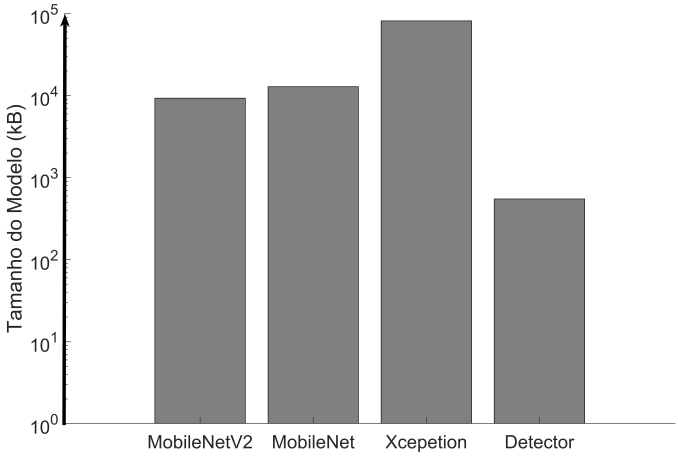



**Figura 8: Comparação entre o tamanho em bytes de diferentes arquiteturas de redes neurais. O tamanho do modelo depende diretamente da quantidade de parâmetros utilizados pela arquitetura.**

no treinamento do aprendizado federado nesse cenário. Por outro lado, no caso não-IID os detectores convergem em $(7 \pm 4)$ para o conjunto de dados CIFAR-10, enquanto $(4 \pm 2)$ para o conjunto de dados MNIST, enquanto a arquitetura MobileNet converge após $(150 \pm 70)$ épocas de treinamento. Isso apresenta uma diferença na comunicação para o treinamento dos modelos de 94% no pior caso.

Por fim, para o conjunto de dados CIFAR-10 no cenário não-IID com duas classes por cliente os detectores convergem em $(21 \pm 18)$ épocas e com o conjunto de dados MNIST são necessárias $(4 \pm 2)$ épocas de treinamento, enquanto o modelo profundo converge após $(70 \pm 30)$ épocas de treinamento. Assim, ao utilizar o modelo um-contra-todos nesse cenário, o sistema apresenta uma redução de bytes enviados durante o treinamento dos modelos de aproximadamente 59,6%. Portanto, o modelo um-contra-todos apresenta uma redução significativa na comunicação para o treinamento de modelos no aprendizado federado, ao passo que a acurácia dos detectores gerados e do modelo final possuem um bom desempenho de classificação, mesmo em distribuições de dados heterogêneas.

## 5. Conclusão e Trabalhos Futuros

Este artigo apresentou o ATHENA-FL, um sistema de aprendizado federado que utiliza o agrupamento de clientes para redução da heterogeneidade dos dados durante o treinamento. Além disso, o sistema aplica o método um-contra-todos para a criação de classificadores genéricos através do compartilhamento de modelos entre grupos. Os resultados mostram que a comunicação durante as épocas de treinamento é eficiente, reduzindo entre 59,6% até 94% a quantidade de bytes transmitidos em comparação a abordagem tradicional com a rede neural MobileNet. A acurácia do modelo um-contra-todos depende do cenário de distribuição de dados utilizado, sendo até 18% maior no melhor caso e apresentando um desempenho melhor do que o modelo tradicional na maioria dos cenários avaliados. Para estabilizar o desempenho dos detectores em diferentes distribuições de dados, uma alternativa é incluir técnicas de aumento do conjunto de dados através da adição de ruído e utilização de conjuntos de dados com maior diversidade de amostras. Outra abordagem possível é ajustar os hiperparâmetros dos detectores utilizados, como arquitetura da rede neural e possivelmente a utilização de detectores específicos para cada classe, conforme a dificuldade em detectar o padrão no conjunto de dados.

Em trabalhos futuros pretende-se estender a proposta para identificar vulnerabilidade em redes de computadores para identificar tipos de ataques e reagir de forma eficaz às ameaças.

## Referências


Blondel, V. D. et al. (2008). Fast Unfolding of Communities in Large Networks. *Journal of Statistical Mechanics: Theory and Experiment*, páginas 1–12.

Chu, D., Jaafar, W. e Yanikomeroglu, H. (2022). On the Design of Communication-Efficient Federated Learning for Health Monitoring. *IEEE GLOBECOM*, páginas 1–6.

de Souza, L. A. C. et al. (2022). Aprendizado Federado com Agrupamento Hierárquico de Clientes para Aumento da Acurácia. Em *SBRC*, páginas 545–558.

Ester, M., Kriegel, H.-P., Sander, J., Xu, X. et al. (1996). A Density-based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. Em *KDD*, páginas 226–231.

Fu, L. et al. (2022). Client Selection in Federated Learning: Principles, Challenges, and Opportunities. *arXiv preprint arXiv:2211.01549*, páginas 1–8.

Krizhevsky, A. et al. (2009). Learning Multiple Layers of Features from Tiny Images.

Lai, F., Zhu, X., Madhyastha, H. V. e Chowdhury, M. (2021). Oort: Efficient Federated Learning via Guided Participant Selection. Em *USENIX OSDI*, páginas 19–35.

LeCun, Y., Cortes, C. e Burges, C. J. (2010). MNIST Handwritten Digit Database. http://yann.lecun.com/exdb/mnist/.

Li, H., Cai, Z., Wang, J., Tang, J., Ding, W., Lin, C.-T. e Shi, Y. (2022). FedTP: Federated Learning by Transformer Personalization. *arXiv preprint arXiv:2211.01572*, páginas 1–14.

Luo, B. et al. (2022). Tackling System and Statistical Heterogeneity for Federated Learning with Adaptive Client Sampling. Em *IEEE INFOCOM*, páginas 1739–1748.

McMahan, B. et al. (2017). Communication-efficient Learning of Deep Networks from Decentralized Data. *Artificial Intelligence and Statistics*, páginas 1273–1282.

Neto, H. N. et al. (2021). FedSA: Arrefecimento Simulado Federado para a Aceleração da Detecção de Intrusão em Ambientes Colaborativos. Em *SBRC*, páginas 280–293.

Ouyang, X. et al. (2021). ClusterFL: a Similarity-Aware Federated Learning System for Human Activity Recognition. Em *Proceedings of the International Conference on Mobile Systems, Applications, and Services*, páginas 54–66.

Tan, A. Z., Yu, H., Cui, L. e Yang, Q. (2022). Towards Personalized Federated Learning. *IEEE Transactions on Neural Networks and Learning Systems*, páginas 1–17.

Wang, H. et al. (2020a). Optimizing Federated Learning on Non-IID Data with Reinforcement Learning. Em *IEEE INFOCOM*, páginas 1698–1707.

Wang, J. et al. (2020b). Tackling the Objective Inconsistency Problem in Heterogeneous Federated Optimization. *NeurIPS*, 33:7611–7623.

Zhong, Z. et al. (2022). FLEE: A Hierarchical Federated Learning Framework for Distributed Deep Neural Network over Cloud, Edge and End Device. *ACM TIST*, páginas 1–24.

Zhu, Y., Markos, C., Zhao, R., Zheng, Y. e James, J. (2021). FedOVA: One-vs-All Training Method for Federated Learning with Non-IID Data. Em *IEEE IJCNN*, páginas 1–7.