COPPE
Instituto Alberto Luiz Coimbra de Pós-Graduação e Pesquisa de Engenharia
UFRJ

TÉCNICAS DE SELEÇÃO DE MODELOS DE APRENDIZADO DE MÁQUINA PARA DETECÇÃO DE DEFEITOS INDUSTRIAIS

Luana Gantert Reis Martinez

Tese de Doutorado apresentada ao Programa de Pós-graduação em Engenharia Elétrica, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Doutor em Engenharia Elétrica.

Orientador: Miguel Elias Mitre Campista

Rio de Janeiro
Abril de 2026

TÉCNICAS DE SELEÇÃO DE MODELOS DE APRENDIZADO DE MÁQUINA PARA DETECÇÃO DE DEFEITOS INDUSTRIAIS

Luana Gantert Reis Martinez

TESE SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR EM CIÊNCIAS EM ENGENHARIA ELÉTRICA.

Orientador: Miguel Elias Mitre Campista

Aprovada por: Prof. Miguel Elias Mitre Campista
Prof. Dianne Scherly Varela de Medeiros
Prof. Célio Vinicius Neves de Albuquerque
Prof. Rodrigo de Souza Couto
Prof. Luís Henrique Maciel Kosmalski Costa

RIO DE JANEIRO, RJ – BRASIL
ABRIL DE 2026

Martinez, Luana Gantert Reis

Técnicas de Seleção de modelos de aprendizado de máquina para detecção de defeitos industriais/Luana Gantert Reis Martinez. – Rio de Janeiro: UFRJ/COPPE, 2026.

XIII, 76 p.: il.; 29, 7cm.

Orientador: Miguel Elias Mitre Campista

Tese (doutorado) – UFRJ/COPPE/Programa de Engenharia Elétrica, 2026.

Referências Bibliográficas: p. 72 – 76.

1. Aprendizado de Máquina. 2. IIoT. 3. Aprendizado Federado. I. Campista, Miguel Elias Mitre. II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia Elétrica. III. Título.

À minha filha Maria Elisa.

Agradecimentos

Agradeço ao meu marido, Renan Martinez, por acreditar na minha capacidade de alcançar meus sonhos mesmo antes do vestibular. Aos meus pais, Iris Gantert e Everaldo Reis, por serem fundamentais na minha jornada. Aos meus familiares Victor Gantert, Gabriel Gantert, Beatriz Parente, Gabriel Reis e Thaiany Moraes pela torcida e carinho.

Agradeço aos colegas do Grupo de Teleinformática e Automação pelo acolhimento e pelas contribuições. Em especial, a Mariana Maciel, Daiane Pereira, Roberto Pacheco e Lucas Gomes que se tornaram meus grandes amigos durante o caminho.

Agradeço à minha equipe da Divisão de Automação e Controle do Grupo de Sistemas Digitais do Instituto de Pesquisas da Marinha por toda a compreensão e apoio neste último ano. Serei eternamente grata por cada dia ao lado de vocês.

Agradeço à Patrícia de Paula, à Yasmin Manhães e à Gabriela Borges pelo direcionamento e cuidado ao longo dos anos. Espero poder retribuir todo o apoio recebido.

Agradeço ao meu orientador, Miguel Elias Mitre Campista, pelos ensinamentos, pela paciência e pelo apoio. Obrigada pelo exemplo e por sempre me motivar com novos desafios. Agradeço aos professores Rodrigo de Souza Couto, Dianne Scherly Varela de Medeiros, Luís Henrique Maciel Kosmalski Costa e Célio Vinicius Neves de Albuquerque por fazerem parte da banca examinadora.

Por fim, agradeço ao apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) e do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq).

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Doutor em Ciências (D.Sc.)

TÉCNICAS DE SELEÇÃO DE MODELOS DE APRENDIZADO DE MÁQUINA PARA DETECÇÃO DE DEFEITOS INDUSTRIAIS

Luana Gantert Reis Martinez

Abril/2026

Orientador: Miguel Elias Mitre Campista

Programa: Engenharia Elétrica

Esta tese propõe métodos de seleção de modelos de aprendizado de máquina, a partir de características do espectro sonoro, para o reconhecimento de falhas do maquinário industrial. São adotados dois conjuntos de dados com amostras de sons no contexto industrial: MIMII e ToyADMOS. As características espectrais descrevem os sons de maneira concisa, exigindo menos poder computacional do que trabalhos que adotam espectrogramas e redes neurais convolucionais. O trabalho de análise e seleção de modelos é realizado através do aprendizado de máquina tanto centralizado quanto federado. Para o aprendizado centralizado, são avaliados os desempenhos do método Super Learner de combinação de algoritmos; os cinco algoritmos utilizados no método separadamente: *AdaBoost*, Naïve Bayes, K-Vizinhos Mais Próximos (*K-Nearest Neighbors – KNN*), Máquina de Vetores de Suporte (*Support Vector Machine – SVM*) e Floresta Aleatória; e duas redes neurais do tipo perceptron de múltiplas camadas. Entretanto, esta análise revela que não é possível identificar um algoritmo que seja o mais indicado para todos os equipamentos analisados. Desta forma, o aprendizado federado possibilita que um novo modelo global seja construído iterativamente com base nos parâmetros treinados localmente pelos diferentes clientes. Nesta tese, cada cliente se conecta a múltiplos servidores e seleciona aquele com o modelo mais promissor. Um método de seleção de modelos é proposto com base nos resultados atuais de cada cliente, uma vez que a análise centralizada revela que, para cada equipamento/cliente, há sempre um modelo que apresenta os melhores resultados. O método pode apresentar uma única rodada de avaliação ou um intervalo de avaliação para seleção de servidor até que cada cliente permaneça treinando um único modelo. Os resultados mais significativos representam um aumento de 55.5% da métrica AUC-ROC do modelo final.

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Doctor of Science (D.Sc.)

MACHINE LEARNING MODEL SELECTION TECHNIQUES FOR INDUSTRIAL DEFECT DETECTION

Luana Gantert Reis Martinez

April/2026

Advisor: Miguel Elias Mitre Campista

Department: Electrical Engineering

This thesis proposes methods for selecting machine learning models based on sound spectrum features for the recognition of industrial machinery failures. Two datasets with industrial sound samples are adopted: MIMII and ToyADMOS. Spectral features describe sounds concisely, requiring less computational power than studies that adopts spectrograms and convolutional neural networks. The analysis and selection of models are conducted using both centralized and federated machine learning approaches

In centralized learning, the performance of the Super Learner method for combining algorithms is evaluated; the five algorithms used in the method, assessed separately: AdaBoost, Naïve Bayes, K-Nearest Neighbors (KNN), Support Vector Machine (SVM), and Random Forest (RF); as well as two multilayer perceptron neural networks. However, this analysis shows that it is not possible to identify a single algorithm that is best suited for all the equipment analyzed. In this way, federated learning enables a new global model to be built iteratively based on parameters trained locally by different clients. In this thesis, each client connects to multiple servers and selects the one with the most promising model. A model selection method is proposed based on the current results of each client, since centralized analysis shows that, for each piece of equipment/client, there is always a model that achieves the best results. The method may involve a single evaluation round or an evaluation interval for server selection, until each client continues training a single model. The most significant results represent a 55.5% increase in the AUC-ROC metric of the final model.

Sumário

Lista de Figuras **x**

Lista de Tabelas **xiii**

1 Introdução **1**

2 Trabalhos Relacionados **4**

2.1 Aprendizado de máquinas no cenário industrial 4

2.2 Detecção de eventos através de características do sonoras 5

2.3 Super learner 6

2.4 Redução de recursos e seleção de clientes no aprendizado federado . . 7

3 Conjuntos de Dados e Características Espectrais **9**

3.1 Conjuntos de dados 9

3.1.1 MIMII 9

3.1.2 ToyADMOS 10

3.2 Características espectrais 11

3.2.1 Domínio da frequência 11

3.2.2 Domínio do tempo 12

3.2.3 Domínio cepstral 12

4 Generalização de modelos para detecção de defeitos industriais **14**

4.1 Etapas para implementação do aprendizado de máquinas 15

4.1.1 Algoritmos de aprendizado de máquina 16

4.1.2 Redes neurais 18

4.1.3 Métricas de avaliação 20

4.1.4 Acurácia 20

4.2 Classificadores binários 23

4.2.1 Primeira fase experimental 23

4.2.2 Segunda fase experimental 29

4.3 Classificadores multiclasse 34

4.3.1 Recursos computacionais para treinamento do modelo 40

4.3.2 Generalização do modelo 40

5 Seleção de modelos baseada em avaliação das federações 44

5.1 Aprendizado federado tradicional 45

5.1.1 Seleção de modelo com parâmetro r fixo 46

5.1.2 Seleção de modelos baseada em regressão linear e redução de latência 48

5.1.3 Seleção de modelo com intervalo de avaliação 48

5.2 Parâmetro β 50

5.3 Experimentos 52

5.4 Resultados 54

5.4.1 Treinamento dos modelos locais 54

5.4.2 Teste dos modelos 58

5.4.3 Análise de decisão dos clientes 58

5.4.4 Latência acumulada 60

5.4.5 Avaliação do modelo final 61

6 Conclusão e Trabalhos Futuros 68

Referências Bibliográficas 72

Lista de Figuras

4.1 Fluxograma com as etapas adotadas. 16
4.2 Método Super Learner para combinação de algoritmos de aprendizado de máquina. 18
4.3 Número de parâmetros treináveis das redes neurais selecionadas através da otimização Bayesiana. 22
4.4 AUC-ROC para o aprendizado de máquinas tradicional. 25
4.5 *Recall* para o aprendizado de máquinas tradicional. 26
4.6 Precisão para o aprendizado de máquinas tradicional. 27
4.7 MCC para o aprendizado de máquinas tradicional. 28
4.8 AUC-ROC e F1-Score para os diferentes algoritmos e as distintas SNR. 32
4.9 AUC-PR e MCC para os diferentes algoritmos e as distintas SNR. . . 33
4.10 AUC-ROC, F1-Score, AUC PR e MCC para o conjunto de dados ToyADMOS. 34
4.11 Classificação binária e modelos multiclasse propostos em busca da generalização da classificação em cenários industriais. 36
4.12 Matrizes de confusão dos algoritmos LGBM e SVM para classificação multiclasse mistura parcial separadas pelas diferentes SNRs. 37
4.13 Matrizes de confusão para os algoritmos LGBM e SVM na classificação multiclasse para o conjunto de dados ToyADMOS. 38
4.14 Matrizes de confusão para os algoritmos LGBM e SVM para a classificação multiclasse com SNRs totalmente misturadas utilizando todas as amostras. 39
4.15 Comparação do tempo de treinamento para os classificadores binários e multiclasse. 40
4.16 AUC-ROC e F1-Score nas três abordagens para o conjunto de dados MIMII. 41
4.17 AUC-ROC e F1-Score para as duas abordagens para o conjunto de dados ToyADMOS. 42

5.1 Aprendizado federado tradicional. 45
5.2 Proposta de avaliação das federações com r fixo. 46

5.3 Procedimento de desligamento de federações na etapa de avaliação com r fixo. 47
5.4 Proposta de avaliação das federações baseada em regressão linear. . . 48
5.5 Proposta de avaliação das federações com intervalo de poda. 49
5.6 Número de federações por clientes considerando a rodada de comunicação rc das propostas. 50
5.7 Memória RAM utilizado durante o treinamento dos modelos para o conjunto de dados MIMII. 55
5.8 Memória RAM utilizada durante o treinamento dos modelos para o conjunto de dados TOYADMOS. 56
5.9 Evolução do tempo de treinamento local dos modelos durante as rodadas de comunicação com os servidores para o conjunto de dados MIMII. 57
5.10 Evolução do tempo de treinamento local dos modelos durante as rodadas de comunicação com os servidores para o conjunto de dados TOYADMOS. 58
5.11 Evolução da função de custo no treinamento dos modelos durante as rodadas de comunicação com os servidores para $rfixo$ e conjunto de dados MIMII. 59
5.12 Evolução da função de custo no treinamento dos modelos durante as rodadas de comunicação com os servidores para rLR e conjunto de dados MIMII. 60
5.13 Evolução da função de custo no treinamento dos modelos durante as rodadas de comunicação com os servidores para $rfixo$ e conjunto de dados TOYADMOS. 61
5.14 Evolução da função de custo no treinamento dos modelos durante as rodadas de comunicação com os servidores para rLR e conjunto de dados TOYADMOS. 62
5.15 Evolução da AUC-ROC dos modelos durante as rodadas de comunicação com os servidores para o conjunto de dados MIMII. 63
5.16 Evolução da AUC-ROC dos modelos durante as rodadas de comunicação com os servidores para o conjunto de dados ToyADMOS. . . . 63
5.17 Escolha dos clientes pelos servidores nas 10 rodadas independentes para o conjunto de dados MIMII. 64
5.18 Escolha dos clientes pelos servidores nas 10 rodadas independentes para o conjunto de dados TOYADMOS. 65
5.19 Escolha dos clientes pelos servidores nas 10 rodadas independentes para o conjunto de dados TOYADMOS. 65

5.20 Latência acumulada durante as rodadas de comunicação de treinamento para os distintos valores de r e β para ambos os conjuntos de dados. 66
5.21 Comparação por equipamento da AUC-ROC obtida pelos modelos finais de aprendizado federado e centralizado. 67

Lista de Tabelas

3.1 Conjunto de dados MIMII. 10
3.2 Conjunto de dados ToyADMOS com razão de amostras anômalas. . . 11

4.1 Parâmetros variáveis na busca de hiperparâmetros 19
4.2 Redes neurais selecionadas para cada máquina individual no conjunto de dados MIMII. 20
4.3 Hiperparâmetros das redes neurais selecionadas para cada máquina individual no conjunto de dados MIMII. 20
4.4 Redes neurais selecionadas para cada máquina individual no conjunto de dados ToyADMOS. 20
4.5 Hiperparâmetros das redes neurais selecionadas para cada máquina individual no conjunto de dados ToyADMOS. 21
4.6 Valores dos hiperparâmetros dos classificadores base. 24
4.7 Hiperparâmetros das redes neurais adotadas para o conjunto de dados MIMII. 24
4.8 Hiperparâmetros das redes neurais adotadas para o conjunto de dados ToyADMOS. 25
4.9 Hiperparâmetros dos algoritmos de aprendizado de máquinas selecionados. 31

5.1 Correspondência entre as máquinas no conjunto de dados original e suas identificações de clientes adotadas no aprendizado federado para o conjunto de dados MIMII. 53
5.2 Correspondência entre as máquinas no conjunto de dados original e suas identificações de clientes adotadas no aprendizado federado para o conjunto de dados ToyADMOS. 54
5.3 Latências usadas na etapa experimental para os diferentes servidores. 54

Capítulo 1

Introdução

As tecnologias baseadas em aprendizado de máquina surgem na indústria 4.0 (I4.0) como uma forma de monitorar os equipamentos industriais, os processos de fabricação e os de logística. Assim, é possível reduzir desperdícios e resíduos, garantir que o maquinário opere de forma segura e de acordo com padrões pré-estabelecidos e aumentar a eficiência energética das operações WANG *et al.* (2018); GHOBAKHLOO (2020). Além disso, essas soluções permitem que as manutenções preditivas e corretivas sejam realizadas assim que padrões de operação diferentes do esperado sejam detectados ACETO *et al.* (2019); LI *et al.* (2018); BOCHIE *et al.* (2021).

Além do tradicional aprendizado de máquina centralizado, o aprendizado federado proposto por MCMAHAN *et al.* (2017) surge como uma técnica distribuída capaz de treinar os modelos sem que dados sensíveis dos clientes sejam compartilhados. Os modelos são treinados pelos clientes com seus dados locais e os parâmetros são enviados ao servidor para que sejam agregados e gerem um modelo global de maneira iterativa. O modelo global é atualizado e redistribuído até que um critério de parada seja atingido. A colaboração entre os clientes durante a etapa de treinamento aumenta a quantidade de dados disponíveis e é capaz de melhorar o desempenho dos modelos finais. Então, o aprendizado federado contribui com aplicações já populares no aprendizado centralizado, como por exemplo, o processamento de áudio e vídeo LI *et al.* (2020); RAMOS *et al.* (2021).

O treinamento distribuído introduz um desafio ao aprendizado federado, que é treinar eficientemente o modelo considerando a heterogeneidade dos clientes e seus dados. As técnicas de seleção de clientes são aplicadas considerando grupos com maior tendência para contribuir com a convergência do modelo global, sendo este grupo selecionado em etapa precedente ou mesmo durante o treinamento FU *et al.* (2023). Essas abordagens assumem que o servidor de aprendizado federado possui informações que permitem selecionar os melhores clientes para o treinamento do modelo e não consideram a possibilidade de existir outra arquitetura ou configuração de modelo mais adequada para determinados clientes. Apesar das aplicações

de soluções baseadas em aprendizado federado em incontáveis áreas, este trabalho considera apenas o contexto industrial e, especificamente, a manutenção reativa dos equipamentos industriais. Trabalhos anteriores exploram a adoção do aprendizado centralizado para detecção de padrões anômalos em equipamentos. Para isso, amostras sonoras do próprio equipamento são usadas como entrada dos modelos de detecção de defeitos. As abordagens podem ser baseadas em características espectrais extraídas dos sons originais NATESHA e GUDDETI (2021); GANTERT *et al.* (2021, 2022), que reduzem a complexidade do modelo para a tarefa de classificação em comparação com abordagens baseadas em espectrogramas SUEFUSA *et al.* (2020); TAO *et al.* (2023). A extração das características espectrais reduz também o tempo de treinamento dos modelos locais no aprendizado federado e, como consequência, o tempo de treinamento do modelo global. Mesmo com a adoção dessas características, uma seleção prévia do modelo mais apropriado para cada equipamento, ou cliente no contexto federado, continua sendo um desafio.

A escolha do melhor modelo para uma aplicação ou conjunto de dados em particular é uma tarefa que, usualmente, não envolve a participação dos clientes, por mais que estes sejam os maiores interessados. Dessa forma, esta tese considera que os clientes são capazes de selecionar o modelo de aprendizado a ser treinado a partir de um conjunto de possibilidades disponíveis e conforme suas próprias métricas de avaliação. Dentre as métricas adotadas para avaliação e seleção dos modelos estão a latência simulada e a métrica AUC-ROC (Área Sob a Curva Característica de Operação do Receptor – (*Area Under the Receiver Operating Characteristic Curve*)). Os conjuntos de dados adotados na etapa experimental são o MIMII (*Malfunctioning Industrial Machine Investigation and Inspection*) PUROHIT *et al.* (2019) e o ToyADMOS KOIZUMI *et al.* (2019), pois contêm amostras sonoras de diferentes tipos de máquinas industriais e de miniaturas de máquinas, respectivamente. Os resultados mostram um aumento da métrica do modelo final comparado com o aprendizado federado tradicional, aquele em que os clientes não podam conexões.

Este trabalho é dividido em duas partes: (i) a detecção de defeitos industriais através de sons a partir de algoritmos de aprendizado de máquina centralizados e (ii) propostas de seleção de modelo para desempenhar a mesma tarefa, mas aplicada no contexto do aprendizado federado. Para o aprendizado centralizado são adotados os algoritmos *AdaBoost*, Naïve Bayes, Máquinas de vetores de suporte (*Support Vector Machine – SVM*), K-Vizinhos Mais Próximos (*K-Nearest Neighbors – KNN*) , Floresta aleatória (*Random Forest – RF*) e o método de combinação desses algoritmos, denominado Super Learner. Além disso, duas arquiteturas distintas de redes neurais são analisadas comparativamente com os demais algoritmos. Já as propostas para a seleção de modelo mais promissor consideram a existência de múltiplos servidores, cada um com um modelo distinto dos demais, e múltiplos clientes. Inicialmente, cada

um dos clientes participa do treinamento de todos os modelos. Então, uma rodada de avaliação é considerada, tendo como principal objetivo avaliar a permanência do cliente no processo de treinamento de cada um dos modelos. Para isso, na etapa de avaliação, os clientes enviam a um nó centralizador o desempenho obtido em cada modelo até então durante o treinamento com seus dados locais. O nó centralizador é chamado de nó comparador, uma vez que sua função é avaliar e manter o cliente no treinamento apenas da federação com modelo mais promissor e se desligar das demais. Dessa forma, o consumo de recursos computacionais é reduzido ao diminuir o número de clientes em cada servidor através da poda. Outro aspecto observado é o impacto na métrica de avaliação ao remover clientes que não contribuem para a convergência do modelo.

O restante deste trabalho está organizado da seguinte forma: no Capítulo 2 são apresentados os trabalhos relacionados; no Capítulo 3 são apresentadas as características espectrais extraídas das amostras originais e o conjunto de dados; o Capítulo 4 apresenta o aprendizado de máquinas tradicional e os resultados para os conjuntos de dados selecionados; o Capítulo 5 apresenta as propostas de seleção de modelos e os resultados obtidos; o Capítulo 6 apresenta a conclusão e os trabalhos futuros.

Capítulo 2

Trabalhos Relacionados

Os trabalhos selecionados nesta seção abordam propostas baseadas em aprendizado de máquinas para detecção de eventos. Assim, a literatura existente está relacionada com a proposta deste trabalho por pelo menos uma das seguintes razões: (a) monitoramento industrial; (b) uso de amostras de som como entrada;(c) poda de parâmetros e camadas em redes neurais no aprendizado federado; e/ou (d) propostas baseadas no método de combinação de modelos Super Learner.

2.1 Aprendizado de máquinas no cenário industrial

GUO *et al.* (2018) propõem um sistema que utiliza uma rede neural convolucional (*Convolutional Neural Network* – CNN) para classificar o estado dos rolamentos a partir de sinais de vibração. O método proposto inclui um módulo para classificação de estado e um módulo de adaptação de domínio para realizar inferências sobre novas amostras não rotuladas através da transferência de aprendizado. VAKARUK *et al.* (2021) utilizam redes de aprendizado profundo para prever defeitos em veículos guiados automaticamente utilizados principalmente na indústria para auxiliar os processos industriais. O artigo propôs uma rede LSTM (*Long Short-Term Memory*) em conjunto com uma rede CNN em um cenário 5G, adotando o tráfego de rede como dados de entrada. Assim, as propostas mencionadas utilizam redes neurais para detecção de falhas, mas não consideram amostras de áudio como entrada nem a generalização de algoritmos para manutenção industrial.

ORRÙ *et al.* (2020) propõem uma abordagem com uma rede MLP (*Multilayer perceptron*) em conjunto com o algoritmo Máquinas de Vetores de Suporte (*Support-Vector Machine* – SVM) para diagnosticar falhas na bomba centrífuga na indústria de óleo e gás. O conjunto de dados inclui dados de temperatura da bobina do motor, vibração dos rolamentos, taxa de fluxo e deslocamento axial. CHAI e ZHAO (2019) adotam o algoritmo de Floresta Aleatória (*Random Forest* – (RF)) com ranqueamento e seleção para análise de características estáticas e dinâmicas. O método

proposto pode detectar a falha no processo clássico *Tennessee Eastman*(BATHELT *et al.* (2015)) que contém elementos como reator, o condensador de produto, o separador de vapor-líquido, entre outros. Dessa forma, ambas as propostas consideram algoritmos supervisionados para a classificação de estados, não adotando amostras de som nem características espectrais.

2.2 Detecção de eventos através de características do sonoras

A literatura aponta a adoção de algoritmos de aprendizado de máquinas para classificação de amostras sonoras em diferentes áreas. BUI *et al.* (2020) combinam CNN com a extração de características e comparam com os algoritmos K-Vizinhos mais Próximos (*K-Nearest Neighbor* – (KNN)), SVM, RF e AdaBoost para classificar a densidade de tráfego em diferentes períodos do dia. A representação visual dos sons urbanos é extraída através dos espectrogramas na escala Mel e MFCCs. MOHAIMENUZZAMAN *et al.* (2022) exploram a classificação de sons ambientes utilizando microcontroladores e adotando espectrogramas e XOR-Networks RASTEGARI *et al.* (2016), um tipo de CNN com pesos e entrada assumindo valores binários. Essas propostas adotam amostras de som e algoritmos de aprendizado supervisionado sem focar no contexto industrial.

Na área de cuidados médicos, PAHAR *et al.* (2021) discriminam casos positivos de coronavírus (COVID-19) utilizando sons de tosse gravados com *smartphones* como entrada para o treinamento de modelos. Características espectrais como MFCCs, Taxa de Cruzamento por Zero (*Zero Crossing Rate* – ZCR) e Kurtosis são adotadas para representar as amostras sonoras. Os autores comparam o desempenho da SVM e classificadores baseados em regressão logística, além de cinco diferentes arquiteturas de redes neurais. JAMIL e ROY (2023) propõem um sistema de detecção de doenças vasculares por meio de fonocardiogramas. As características são selecionadas por métodos bioinspirados como algoritmos genéticos (*Genetic Algorithm* – GA) e otimização por enxame de partículas (*Particle Swarm Optimization* – PSO). Os algoritmos KNN, SVM e árvores de decisão foram adotados para a classificação.

Ao invés da adoção das características espectrais, determinados trabalhos convertem as amostras sonoras em espectrogramas para que ferramentas usuais em processamento de imagens sejam utilizadas. SUEFUSA *et al.* (2020) realizam experiências utilizando o conjunto de dados MIMII com uma rede *autoencoder* e um *autoencoder* variacional, usando espectrogramas em escala Mel como entrada. A proposta de TAMA *et al.* (2022) adota como entrada imagens de espectrogramas

em escala Mel extraídas do conjunto de dados MIMII e a rede EfficientNet para detectar defeitos. Os autores aplicam técnicas de sobreamostragem para aumentar sinteticamente o número de amostras e lidar com a natureza desequilibrada do conjunto de dados. Em ambas as propostas, os autores utilizam espectrogramas de Mel como entrada, uma vez que o processamento de imagens é outra alternativa para o processamento de áudio. Essas abordagens geralmente adotam redes neurais convolucionais e, consequentemente, precisam treinar mais parâmetros do que as propostas baseadas em caraterísticas espectrais.

NATESHA e GUDDETI (2021) propõem um sistema de monitoramento e classificação de comportamento anômalo através das amostras sonoras, adotando o conjunto de dados MIMII. Dessa forma, eles utilizam os coeficientes Mel-Cepstrais e os coeficientes de predição linear como entrada dos modelos centralizados. Redes neurais de múltiplas camadas são adotadas, i.e., os algoritmos RF, Regressão Logística (*Logistic Regression* – LR), SVM e *AdaBoost* (AB). Como resultado, as redes neurais de múltiplas camadas e o algoritmo AB apresentam resultados superiores nas métricas AUC-ROC e F1-Score.

Os trabalhos de detecção de anomalias baseados em sons industriais têm como ponto comum o desconhecimento do melhor modelo a ser adotado na prática. Isso faz com que esses trabalhos realizem um procedimento de avaliação de múltiplos modelos para identificação daquele que possui o melhor desempenho. Os resultados são tipicamente semelhantes, o que levanta questões sobre a validade em cenários reais que possivelmente apresentam características diferentes.

Nesse sentido, o trabalho atual adota o método Super Learner para a classificação de amostras anômalas em comparação com outros algoritmos. Além disso, o desempenho de redes neurais é avaliado considerando a seleção de modelos de aprendizado federado através dos resultados intermediários obtidos pelos clientes durante o treinamento.

2.3 Super learner

Na literatura, HEDEYA *et al.* (2020) adotam Super Learner para melhorar o desempenho de redes neurais profundas em classificação de tipos de veículos em quadros capturados por câmeras de vigilância. Essa proposta combina as redes convolucionais: Resnet 50, Xception, e DenseNet. WEI *et al.* (2022) adota Super Learner para estimar a emissão de gases de efeito estufa em veículos movidos a diesel. Ao ser comparado com sistemas já populares de predição de emissão de gás, a raiz do erro médio quadrático é reduzida em 50%. Dessa forma, trabalhos anteriores também comprovam que a combinação de algoritmos por meio do Super Learner alcança valores de AUC-ROC superiores aos dos algoritmos usados individualmente para a

classificação de cenários não apenas industrial, mas também na detecção de sons urbanos (GANTERT *et al.* (2022)).

Esta tese aplica o algoritmo Super Learner na classificação binária de detecção de defeitos industriais, no contexto da investigação de algoritmos voltados à generalização dos modelos. O Super Learner é comparado com os algoritmos base que o compõem e com redes neurais selecionadas. O desempenho é avaliado através das métricas AUC-ROC, precisão, MCC e *recall*.

2.4 Redução de recursos e seleção de clientes no aprendizado federado

JIANG *et al.* (2022) propõem o PruneFL para, de forma adaptativa, podar os parâmetros e adaptar o tamanho dos modelos durante o aprendizado para minimizar o tempo total de treinamento e reduzir os requisitos computacionais necessários. O sistema inicia selecionando um único cliente para podar o modelo com seus dados locais. A poda adaptativa ocorre, pois o modelo original é ajustado iterativamente, removendo parâmetros ou adicionando-os novamente conforme o procedimento de poda é executado.

LIN *et al.* (2022) realizam o método de poda de parâmetros redundantes. Além disso, os autores consideram o impacto das diferentes camadas da rede neural na métrica de avaliação e, dessa forma, adotam uma heurística para determinar o nível de esparsidade das camadas. Os modelos podados alcançam resultados semelhantes aos modelos completos nos experimentos em aplicações de reconhecimento automático.

O FedNets ALHALABI *et al.* (2023) gera modelos podados da rede original ResNetV2, adotando valores distintos de poda de hiperparâmetros para garantir a diversidade entre os modelos. Os clientes selecionam aleatoriamente N modelos desse conjunto, os treinam com seus dados locais e os combinam através de métodos *ensemble* para gerar um único modelo por cliente. O modelo global permanece sendo o ResNetV2, que é atualizado e pode ter seus parâmetros podados conforme o desempenho dos clientes.

HUANG *et al.* (2022) é proposto um *framework* que realiza treinamento híbrido local e centralizado, no qual os clientes podem enviar seus dados para os servidores federados ou prosseguir com treinamento local. Os autores utilizam um algoritmo de otimização para selecionar os clientes que participarão dos diferentes tipos de treinamento, os parâmetros do aprendizado federado e a alocação dos recursos de banda com o objetivo de minimizar a latência global do aprendizado federado.

Em HU *et al.* (2025) propõe a seleção de clientes baseada em gradiente como forma de contornar as baixas acurácia e velocidade de convergência do modelo global.

O algoritmo proposto, FedGSCS (*Federated Learning based on Gradient Similarity*), torna-se uma opção ao algoritmo *Federated Averaging* (*FedAvg*). O FedGSCS realiza a média dos gradientes obtidos pelos modelos locais e compara o gradiente dos clientes com a média através da métrica de similaridade de cosseno. Os resultados obtidos indicam se determinado cliente auxilia ou não a convergência do modelo global. Desta forma, os autores ranqueiam os resultados e selecionam os melhores clientes. Os resultados utilizando o conjunto de dados MNIST, FMNIST e Shakespeare alcançam redução de até 80% de rodadas de comunicação em determinados cenários, além de aumento da acurácia acima de 16%. Compara-se com os algoritmos FedAvg, Power-of-Choice e DivFL.

De forma semelhante, TRINDADE e DA FONSECA (2026) propõem a seleção de clientes através do ranqueamento. Entretanto, os clientes são inicialmente agrupados com base em seus recursos e características de seus dados ao se conectarem com cada servidor. Por meio de uma função multicritério, a pontuação do grupo é calculada para avaliar o impacto de sua contribuição no treinamento do modelo global. Os clientes do grupo com maior pontuação são selecionados para prosseguir no treinamento. O algoritmo utiliza métricas como utilização de CPU, memória e latência de comunicação, por exemplo, na formulação do problema de otimização. Os resultados experimentais utilizam os conjuntos de dados CIFAR-10 e MNIST e reduz em pelo menos 40% a utilização de recursos comparados a outros métodos baseados em clusterização, sem grade comprometimento na acurácia.

Este trabalho não realiza procedimentos de poda sobre modelos pré-definidos, uma vez que o proposto nesta tese como poda de conexões não foi encontrado anteriormente na literatura. A ideia é selecionar um entre múltiplos modelos com arquitetura ou configurações potencialmente diferentes durante o treinamento federado. A poda consiste, portanto, na remoção dos modelos que apresentam desempenho considerado inferior. Uma poda posterior realizada como a dos trabalhos relacionados desta seção pode ser realizada de forma complementar. Além disso, a seleção do modelo mais adequado para o conjunto de dados parte de cada cliente considerando que este é o maior interessado ao invés de uma análise de resultados dos servidores.

Capítulo 3

Conjuntos de Dados e Características Espectrais

Este capítulo apresenta os conjuntos de dados utilizados nos experimentos de aprendizado centralizado e federado, com detalhes na quantidade de amostras e equipamentos. Além disso, o capítulo apresenta as características espectrais extraídas das amostras sonoras originais.

3.1 Conjuntos de dados

Para a etapa experimental, dois conjuntos de dados que contêm amostras de áudio de equipamentos industriais e de miniaturas de equipamentos são adotados. Esses conjuntos apresentam estados de operação normal, isto é, dentro do padrão esperado, e também amostras que saem do padrão e são consideradas anômalas.

3.1.1 MIMII

O conjunto de dados MIMII (*Malfunctioning Industrial Machine Investigation and Inspection*) (PUROHIT *et al.* (2019)) contém segmentos de áudio de 10 segundos para quatro tipos de equipamento, como mostra a Tabela 3.1. O ID corresponde à identificação de uma máquina específica daquele determinado equipamento. Para cada tipo de equipamento, o conjunto de dados original contém amostras sonoras de 4 máquinas individuais. Agrupando essas amostras por tipo de equipamento, o número de amostras anômalas por equipamento varia entre $2,5\%$ e $8,2\%$, relativamente ao número total de amostras no conjunto de dados. Assim, é possível observar como o conjunto de dados é desequilibrado. Em números totais, tem-se 4075 amostras normais e 1475 anormais para ventiladores industriais, 3749 amostras normais e 456 anormais para bombas hidráulicas, 3204 amostras normais e 890 anormais para trilhos deslizantes, e 3691 amostras normais e 479 anormais para válvulas. Pode-se

observar uma variação no número de amostras conforme o tipo de máquina, sendo que há mais informações disponíveis para ventiladores. Os autores adicionaram ruídos aos áudios originais gravados do conjunto de dados, produzindo três níveis diferentes de relação sinal-ruído (*Signal-to-Noise Ratio* – SNR): 6 dB, 0 dB e −6 dB. Dessa forma, quanto menor a SNR, mais desafiadora é a classificação correta das amostras.

Tabela 3.1: Conjunto de dados MIMII.

Tipo de equipamento	**ID da máquina**	**Amostras normais**	**Amostras anômalas**	**Razão anômalas/total**
Ventiladores	00	1011	407	0,29
	02	1016	359	0,26
	04	1033	348	0,25
	06	1015	361	0,26
	Total	**4075**	**1475**	**0,27**
Bombas Hidráulicas	00	1006	143	0,12
	02	1005	111	0,10
	04	702	100	0,12
	06	1036	102	0,09
	Total	**3749**	**456**	**0,11**
Trilho Deslizante	00	1068	356	0,25
	02	1068	267	0,20
	04	534	178	0,25
	06	534	89	0,14
	Total	**3204**	**890**	**0,22**
Válvulas	00	991	119	0,11
	02	708	120	0,15
	04	1000	120	0,11
	06	992	120	0,11
	Total	**3691**	**479**	**0,12**

3.1.2 ToyADMOS

O conjunto de dados ToyADMOS (KOIZUMI *et al.* (2019)) contém segmentos de áudio de 10 segundos e 10 minutos para três equipamentos em miniatura: ToyCar, ToyConveyor e ToyTrain. Para as etapas experimentais, apenas as amostras de 10 segundos foram utilizadas, uma vez que essa duração é equivalente ao das amostras do conjunto de dados MIMII. A Tabela 3.2 apresenta o número total de amostras utilizadas nas experiências ToyADMOS. A coluna "Número do Caso" é o equivalente ao "ID" no conjunto de dados MIMII e diferencia as máquinas individuais uma das outras. O número de amostras anômalas varia entre $5,4\%$ e $5,8\%$ relativamente ao número total de amostras no conjunto de dados. Além disso, tem-se 21600 amostras normais e 4236 anômalas para ToyCar, 21596 amostras normais e 4440 anômalas para ToyConveyor, e 21600 amostras normais e 4320 anômalas para ToyTrain. Observa-se, então, que o conjunto de dados ToyADMOS fornece mais amostras para cada tipo de equipamento do que o conjunto de dados MIMII, e que o número de

amostras de cada tipo tem baixa variação. Mais uma vez, há mais amostras normais do que anormais.

Tabela 3.2: Conjunto de dados ToyADMOS com razão de amostras anômalas.

Tipo de Equipamento	Caso	Amostras normais	Amostras anômalas	Razão anômalas/total
ToyCar	1	5400	1056	0,16
	2	5400	1060	0,16
	3	5400	1060	0,16
	4	5400	1060	0,16
	Total	**21600**	**4236**	**0,16**
ToyConveyor	1	7200	1600	0,18
	2	7200	1420	0,16
	3	7196	1420	0,16
	Total	**21596**	**4440**	**0,17**
ToyTrain	1	5400	1080	0,17
	2	5400	1080	0,17
	3	5400	1080	0,17
	4	5400	1080	0,17
	Total	**21600**	**4320**	**0,17**

3.2 Características espectrais

As características espectrais adotadas são consideradas alternativas aos espectrogramas e comumente abordados na literatura da área de processamento de sinais acústicos CHAKI (2021). Essas características representam os sinais de áudio originais mais concisamente. As características selecionadas são categorizadas conforme a descrição do som no domínio do tempo, no domínio cepstral e no domínio da frequência. Estas últimas são obtidas convertendo as características no domínio do tempo utilizando, por exemplo, transformada de Fourier.

As características espectrais evitam a necessidade de algoritmos de reconhecimento de imagens para a classificação das amostras e são as mesmas adotadas em trabalhos anteriores GANTERT *et al.* (2022, 2024); GANTERT e CAMPISTA (2024). As características adotadas são apresentadas a seguir:

3.2.1 Domínio da frequência

Centroide espectral (*Spectral centroid* – SC)

O Centroide Espectral descreve a magnitude do centro de massa do espectro. Essa característica está associada à ideia subjetiva de clareza do som. Os sinais com frequências mais altas tendem a ser percebidos como sinais mais claros. Na Equação 3.1, o centroide espectral é a média das frequências presentes nos quadros do sinal ponderadas pelo somatório das suas amplitudes.

$$SC = \frac{\sum_{k=1}^{M} kS(k)}{\sum_{k=1}^{M} S(k)}, \tag{3.1}$$

onde $S(k)$ é a magnitude do espectro no quadro de frequência k.

Largura de banda espectral (*Spectral bandwidth* – SB)

A Largura de Banda Espectral é calculada utilizando o Centroide Espectral SC e é definida como descrita na Equação 3.2.

$$SB = \sqrt{\sum_{k=1}^{M} S(k)(f(k) - SC)^2}. \tag{3.2}$$

Decaimento espectral (*Spectral roll-off* – SR)

O decaimento espectral é a frequência abaixo da qual está 85% da energia do espectro. O SR é obtido através da Equação 3.3.

$$\sum_{k=1}^{SR} S(k) = 0.85 \sum_{k=1}^{M} S(k). \tag{3.3}$$

3.2.2 Domínio do tempo

Taxa de cruzamento por zero (*Zero crossing rate* – ZCR)

A Taxa de Cruzamento por Zero é o número de vezes que um sinal de onda de áudio cruza o zero e é definido da seguinte forma:

$$ZCR = \frac{1}{2N} \sum_{t=0}^{N-2} |sgn(x(t+1)) - sgn(x(t))|, \tag{3.4}$$

onde $sgn(x(t)) = \begin{cases} 1, & \text{if } x(t) \geq 0 \\ -1, & \text{if } x(t) < 0 \end{cases}$.

3.2.3 Domínio cepstral

Coeficientes Mel-cepstrais (*Mel-frequency cepstral coefficients* – MFCCs)

Os coeficientes podem descrever foneticamente o envelope do som. O envelope é a amplitude instantânea de um sinal oscilatório. Essa característica é amplamente adotada em reconhecimento de voz, sendo computada como a operação inversa da transformada discreta de cosseno (*Discrete Cosine Transform* – DCT) da energia cepstral em cada frequência Mel ZHENG *et al.* (2001). Os MFCCs são definidos como se segue:

$$MFCC(c) = \sum_{m=1}^{M_{mfcc}} X_m \cos\left[c(m - \frac{1}{2})\frac{\pi}{M_{mfcc}}\right], c = 1, ..., M_{mfcc}, \tag{3.5}$$

onde M_{mfcc} é o número de coeficientes adotados (no trabalho atual, $M_{mfcc} = 20$), X_m é o log da energia do sinal na m-ésima banda de frequência Mel, e c é o índice do coeficiente cepstral.

A conversão da frequência para a escala de frequência Mel é realizada da seguinte forma:

$$Mel(k) = 2595 \log_{10}\left(1 + \frac{k}{100}\right). \tag{3.6}$$

Capítulo 4

Generalização de modelos para detecção de defeitos industriais

O aprendizado de máquina tradicional consiste no treinamento do modelo de forma centralizada, em que cada equipamento ou cliente possui um modelo específico treinado apenas com seus dados. Ao considerar a classificação binária, cada cliente treina um modelo próprio, gerando uma quantidade de modelos treinados igual ao número de clientes existentes. Em busca de aumentar o poder de generalização dos modelos, existem duas abordagens possíveis: a adoção da classificação multiclasse ou a adoção de modelos binários no contexto do aprendizado federado. Considera-se como poder de generalização a capacidade de um único modelo adequar-se a diferentes tipos de máquinas. As propostas de classificação binária e multiclasse centralizadas contrapõem-se ao aprendizado federado, que consiste em um treinamento colaborativo entre os clientes para obter um modelo global. Sendo assim, os clientes colaboram para que um único modelo possa ser gerado. Tanto a proposta multiclasse centralizada quanto o modelo binário de aprendizado federado consideram o modelo binário centralizado como base para avaliar a qualidade dos modelos.

Além da classificação quanto a forma com que os modelos são treinados, é possível classificar os tipos de tarefas em supervisionadas ou não supervisionadas. No aprendizado supervisionado, é necessário que o conjunto de dados forneça amostras rotuladas para que os modelos sejam treinados. Por outro lado, o aprendizado não supervisionado busca obter padrões entre os dados sem que seja necessário a presença de rótulos.

Este capítulo apresenta o processo de generalização de modelos para detecção de defeitos industriais considerando o aprendizado de máquinas tradicional. As etapas utilizadas na implementação do aprendizado tradicional para tarefas supervisionadas são adotadas tanto para classificação binária quanto da multiclasse. As etapas são: extração das características, divisão do conjunto de dados em treinamento e teste, aplicação de sobreamostragem no conjunto de treinamento, transformação dos

dados, treinamento e avaliação dos modelos. Essas etapas também são adotadas no aprendizado federado. Em seguida, são discutidos os principais algoritmos e redes neurais utilizados, bem como as métricas de desempenho. O capítulo detalha os experimentos realizados com classificadores binários, divididos em uma primeira e segunda fase experimental, e posteriormente explora os classificadores multiclasse, destacando os recursos computacionais empregados e a generalização dos modelos obtidos.

A seleção de modelos para detecção de defeitos industriais consiste em duas etapas experimentais: (a) a primeira é abordada neste capítulo para o aprendizado de máquinas centralizado; (b) a segunda é apresentada no próximo capítulo no cenário de aprendizado federado. Este capítulo apresenta os algoritmos selecionados; o conceito de combinação de algoritmos na busca de obter um modelo final mais robusto; a implementação da classificação multiclasse com base nos modelos utilizados na classificação binária e os resultados obtidos em ambas as propostas centralizadas. Esses resultados também são utilizados para a seleção das redes neurais adotadas no Capítulo 5. A combinação de algoritmos se mostra útil, uma vez que produz respostas mais robustas a partir da saída dos algoritmos base. Desta forma, não é necessário obter o melhor algoritmo de cada equipamento ou cliente.

4.1 Etapas para implementação do aprendizado de máquinas

A Figura 4.1 apresenta um fluxograma com as etapas utilizadas na implementação, neste trabalho, das soluções de aprendizado de máquina para a classificação de amostras sonoras. Com exceção das etapas de treinamento e avaliação dos modelos, as demais também são implementadas desta maneira nas propostas federadas. Inicialmente, as características apresentadas anteriormente são extraídas das amostras sonoras originais. Após a extração das características, um conjunto de dados convertido é gerado em formato tabular, em que cada arquivo .csv contém informação de estado de uma máquina individual do conjunto de dados original.

Em seguida, o conjunto de dados é dividido em dados de treinamento e teste na proporção de 60% e 40%, respectivamente. Uma vez que os dados são desbalanceados, o conjunto de treinamento passa pelo algoritmo de balanceamento de dados SMOTE (*Synthetic Minority Oversampling Technique*) – Técnica de Sobreamostragem Minoritária Sintética) CHAWLA *et al.* (2002). O emprego do SMOTE permite igualar o número de amostras normais e anômalas, pois o algoritmo seleciona uma amostra da classe minoritária e seus N vizinhos mais próximos. Uma das amostras vizinhas é selecionada aleatoriamente e uma amostra sintética é criada no espaço de

características entre a amostra inicialmente selecionada da classe minoritária e seu vizinho.

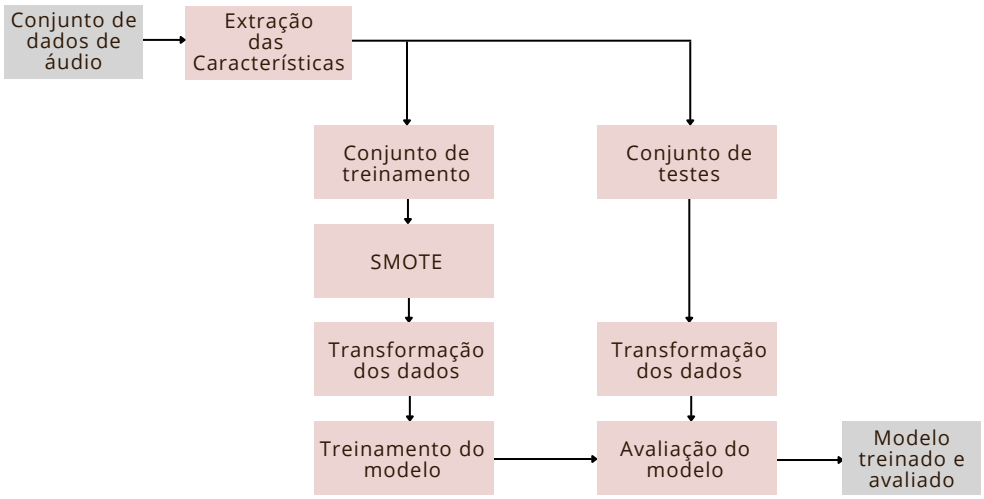


Figura 4.1: Fluxograma com as etapas adotadas.

A transformação dos dados é aplicada separadamente nos conjuntos de treinamento e teste. Essa etapa consiste em escalar os dados através do desvio padrão e da média. Dessa forma, o desvio padrão é igual a 1 e a média igual a 0 nos conjuntos de dados após a transformação SUJON *et al.* (2024). Essa padronização auxilia o modelo a reconhecer os dados e evitar vieses. Além disso, na etapa de transformação, os rótulos são convertidos em valores binários para que a classificação binária possa ser realizada. As amostras da classe anômala assumem valor 1 e, as da classe normal, valor 0. Essa decisão é baseada na noção de que um estado anômalo deve ser considerado pelo operador como um alarme.

Após a etapa de transformação, o modelo é treinado com os dados do conjunto de treinamento e, por fim, testado com dados inéditos do conjunto de teste. Para isso, são utilizadas métricas de avaliação para medir a qualidade do modelo final.

4.1.1 Algoritmos de aprendizado de máquina

Os modelos de aprendizado de máquina são baseados em algoritmos que buscam encontrar padrões e estabelecer regras baseadas nos dados de entrada para dados não conhecidos previamente.

Super Learner

O Super Learner é um algoritmo proposto por Van der Laan *et al.* e consiste em um método para combinação de classificadores VAN DER LAAN *et al.* (2007). Em métodos de combinação de algoritmos, é usual utilizar classificadores base para

construir um modelo com melhores resultados. No caso específico do Super Learner, esses classificadores são apresentados a seguir.

- **AdaBoost (AB):** Neste algoritmo, classificadores são treinados de forma sequencial, em que os pesos atribuídos às classificações incorretas dos anteriores são ajustados. Dessa forma, ao longo do treinamento, os classificadores são capazes de lidar com as amostras mais desafiadoras. Neste trabalho, os classificadores utilizados no AB são árvores de decisão.

- **Naïve Bayes (NB):** Os classificadores do tipo Naïve Bayes são baseados no Teorema de Bayes e assumem que as características das amostras são condicionalmente independentes entre si, dada a classe. Considerando que existem n_c classes e cada amostra do conjunto de treino possui m atributos, as probabilidades de uma amostra inédita x pertencer a cada uma das n_c classes são calculadas assumindo que os m atributos são independentes entre si. Assim, é possível observar como o conjunto de dados é desequilibrado. Dessa maneira, o rótulo de saída é definido de acordo com a maior probabilidade calculada.

- **Máquina de vetores de suporte (*Support Vector Machine* – SVM):** Nesse classificador, os dados são separados através da criação de hiperplanos. A distância mínima entre as classes é denominada margem e o objetivo do algoritmo é encontrar o hiperplano capaz de maximizá-la.

- **K-vizinhos mais próximos (*K-nearest neighbor* – KNN):** Esse classificador considera o número de K amostras mais próximas para rotular uma nova amostra. Essa classe é definida conforme a maioria das classes entre as amostras selecionadas.

- **Floresta Aleatória (*Random Forest* – RF):** O método de floresta aleatória é comumente aplicado em tarefas de regressão e classificação, apesar de também existirem trabalhos que o adotam para eleição de atributos THAKUR e BISWAS (2023). São utilizadas múltiplas árvores de decisão e, no caso de trabalhos de classificação, o rótulo de saída é a moda entre as saídas das árvores individuais.

O Super Learner utiliza validação cruzada K-fold para determinar a melhor combinação de algoritmos para predição ou classificação, sendo este último algoritmo o adotado neste trabalho. A Figura 4.2 apresenta o funcionamento do algoritmo e os classificadores base adotados. No Passo 1, o conjunto de dados é dividido em K blocos, onde cada bloco tem K partes. Uma das partes de K é separada para validação, enquanto os demais blocos são usados para treinar os classificadores base. No Passo 2, os modelos são treinados e validados até que todas as K partes dos

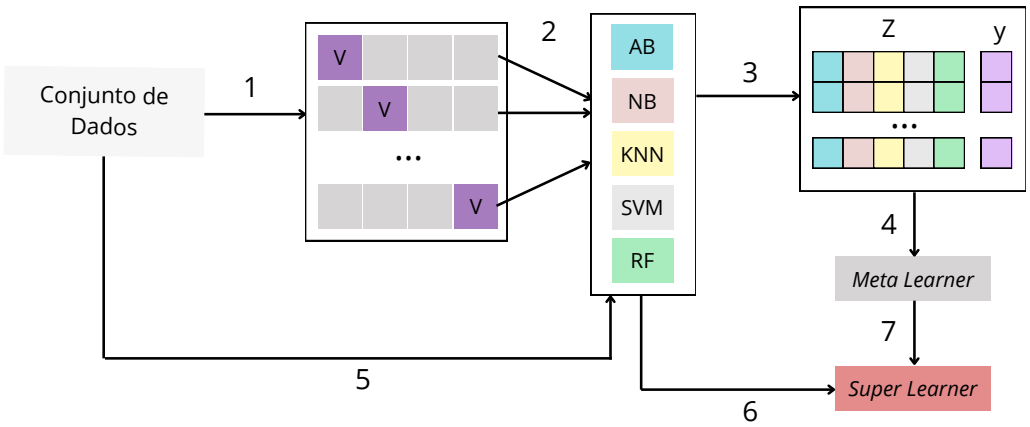


Figura 4.2: Método Super Learner para combinação de algoritmos de aprendizado de máquina.

blocos sejam utilizadas. No Passo 3, uma matriz é construída com todas as Z saídas dos classificadores base. A saída de cada classificador base é ocupada por uma coluna, assim como a figura ilustra. Nesse caso, o algoritmo AB representado pela cor azul ocupará sempre a primeira coluna, o algoritmo NB representado pela cor rosa a segunda coluna e assim sucessivamente. Além das saídas dos classificadores, o valor real y da saída é armazenado na matriz. No Passo 4, a matriz se torna a entrada de outro algoritmo denominado aqui de *meta learner*. Esse algoritmo é treinado com Z para encontrar os pesos ótimos dos classificadores base para alcançar o valor y.

Este trabalho seleciona as árvores de decisão como *meta learner*. No Passo 5, os algoritmos base são treinados com todo o conjunto de dados. O Passo 6 representa as saídas obtidas no passo anterior. Por fim, no Passo 7, o Super Learner é construído aplicando os pesos selecionados pelo *meta learner*. Uma vez que as características do som são adotadas ao invés de espectrogramas, os algoritmos de reconhecimento de imagem podem ser substituídos por algoritmos que necessitem de menos poder computacional.

4.1.2 Redes neurais

As redes neurais buscam tomar decisões através de um modelo matemático inspirado na estrutura do cérebro humano. Neste trabalho, as redes neurais adotadas são do tipo Perceptron de Múltiplas Camadas (*Multilayer Perceptron*). Nesse tipo de arquitetura existem uma ou mais camadas ocultas de neurônios, além das camadas de entrada e saída.

Otimização de parâmetros e hiperparâmetros

Para a escolha das arquiteturas das redes neurais é realizada a otimização dos hiperparâmetros utilizando otimização Bayesiana. A Tabela 4.1 apresenta os parâmetros variados durante a busca. A camada de entrada considera o formato dos dados de entrada, e a camada de saída é composta por um único neurônio com a função de ativação Sigmóide. A otimização é implementada através da biblioteca KerasTuner.

Tabela 4.1: Parâmetros variáveis na busca de hiperparâmetros

Parâmetro	**Intervalo**	**Passo**
Número de camadas ocultas	1 – 5	1
Número de neurônios por camada	32 – 512	32
Taxa de dropout por camada	0,0 – 0,5	0,1
Função de ativação	ReLU	–

Seleção de arquiteturas de redes neurais

As redes neurais obtidas e seus hiperparâmetros para cada equipamento individual do conjunto de dados MIMII são apresentados nas Tabelas 4.2 e 4.3. Uma vez que, para cada equipamento individual, uma arquitetura de rede demonstrou ser mais adequada, os extremos são selecionados considerando o número de parâmetros treináveis. O mesmo ocorre com o conjunto de dados ToyADMOS. Suas redes neurais e seus hiperparâmetros são apresentados nas Tabelas 4.4 e 4.5. A Figura 4.3 representa esses resultados e os relaciona com os respectivos valores da métrica AUC-ROC obtidos na otimização de hiperparâmetros. O eixo x representa o número de camadas ocultas do modelo, e o eixo y, a quantidade de parâmetros treináveis. Quanto maior e mais escuro o círculo, maior o valor da métrica. É possível observar que aumentar o número de parâmetros treináveis ou de camadas ocultas não é o único indicador para obter valores mais elevados de métricas, uma vez que cada equipamento e máquina possui seu próprio grau de dificuldade de classificação.

Tabela 4.2: Redes neurais selecionadas para cada máquina individual no conjunto de dados MIMII.

Máquina	ID	Função de Custo	AUC-ROC	camadas	parametros_treinaveis	learning_rate
Ventilador	00	0,336	0,929	4	189441	0,01
	02	0,098	0,998	1	8321	0,01
	04	0,135	0,981	1	13313	0,001
	06	0,627	1,0	4	176033	0,0001
Trilhos Deslizantes	00	0,005	1,0	3	78977	0,001
	02	9,939	0,593	5	415201	0,001
	04	0,088	0,995	1	12481	0,01
	06	0,112	0,981	1	9985	0,001
Válvulas	00	0,477	0,733	5	263745	0,001
	02	0,109	0,994	1	13313	0,01
	04	0,923	0,63	4	341025	0,01
	06	0,12	0,99	4	103777	0,01
Bombas Hidráulicas	00	5,161	0,761	5	70753	0,01
	02	0,538	0,94	5	205697	0,01
	04	0,0	1,0	3	24833	0,01
	06	0,038	1,0	1	7489	0,01

Tabela 4.3: Hiperparâmetros das redes neurais selecionadas para cada máquina individual no conjunto de dados MIMII.

Máquina	ID	Camada 1		Camada 2		Camada 3		Camada 4		Camada 5	
		neur.	Dropout	neur.	Dropout	neur.	Dropout	neur.	Dropout	neur.	Dropout
Ventilador	00	32	0,4	96	0,2	384	0,3	384	0,4	-	-
	02	320	0,3	-	-	-	-	-	-	-	-
	04	512	0,0	-	-	-	-	-	-	-	-
	06	448	0,4	320	0,4	32	0,4	320	0,4	-	-
Trilhos Deslizantes	00	192	0,2	256	0,2	96	0,0	-	-	-	-
	02	320	0,0	384	0,4	256	0,0	288	0,3	384	0,3
	04	480	0,0	-	-	-	-	-	-	-	-
	06	384	0,4	-	-	-	-	-	-	-	-
Válvulas	00	288	0,0	32	0,4	512	0,0	256	0.2	384	0.0
	02	512	0,0	-	-	-	-	-	-	-	-
	04	64	0,3	480	0,1	384	0,0	320	0,1	-	-
	06	192	0,3	352	0,4	64	0,2	128	0,4	-	-
Bombas Hidráulicas	00	512	0,1	32	0,0	32	0,0	416	0,0	64	0.3
	02	512	0,0	160	0,2	512	0,2	32	0,4	352	0,1
	04	416	0,4	32	0,0	32	0,0	-	-	-	-
	06	288	0,4	-	-	-	-	-	-	-	-

Tabela 4.4: Redes neurais selecionadas para cada máquina individual no conjunto de dados ToyADMOS.

Máquina	ID	Função de Custo	AUC-ROC	n_camadas	parametros_treinaveis
ToyTrain	1	0,129	0,992	1	833
ToyTrain	2	0,468	0,977	3	210817
ToyTrain	3	0,463	0,945	3	30305
ToyTrain	4	0,002	1,0	1	11649
ToyConveyor	1	0,585	0,944	5	153665
ToyConveyor	2	0,008	0,999	5	54625
ToyConveyor	3	0,264	0,986	2	17121
ToyCar	1	2,766	0,829	4	60705
ToyCar	2	1,586	0,835	5	520993
ToyCar	3	2,094	0,878	5	327201
ToyCar	4	0,924	0,972	5	198945

4.1.3 Métricas de avaliação

4.1.4 Acurácia

A acurácia descreve a relação das classificações corretas, os Verdadeiros Positivos (VP) e Verdadeiros Negativos (VN), e todas as classificações realizadas, que incluem

Tabela 4.5: Hiperparâmetros das redes neurais selecionadas para cada máquina individual no conjunto de dados ToyADMOS.

Máquina	ID	Camada 1		Camada 2		Camada 3		Camada 4		Camada 5	
		neur.	Dropout	neur.	Dropout	neur.	Dropout	neur.	Dropout	neur.	Dropout
ToyTrain	case1	32	0,4	-	-	-	-	-	-	-	-
	case2	192	0,0	320	0,4	448	0,2	-	-	-	-
	case3	512	0,4	32	0,1	32	0,3	-	-	-	-
	case4	448	0,4	-	-	-	-	-	-	-	-
ToyConveyor	case1	32	0,4	256	0,0	32	0,4	384	0,0	320	0,4
	case2	32	0,4	32	0,3	384	0,0	96	0,2	32	0,3
	case3	32	0,4	480	0,0	-	-	-	-	-	-
ToyCar	case1	160	0,4	160	0,0	160	0,1	32	0,0	-	-
	case2	32	0,2	352	0,4	512	0,0	352	0,3	416	0,3
	case3	512	0,0	320	0,4	160	0,0	512	0,4	32	0,0
	case4	384	0,3	160	0,1	160	0,0	352	0,2	128	0,1

as classificações incorretas: Falsos Negativos (FN) e Falsos Positivos (FP). Desta forma, essa métrica pode ser interpretada como a taxa de acertos do modelo. Entretanto, na classificação de estados anômalos a acurácia pode mascarar um sistema com taxas significativas de FP e FN.

Precisão

A precisão avalia a taxa de VP entre todos os valores positivos preditos, sendo descrita pela Equação 4.1. Enquanto a *recall* reduz em situações em que o classificador não interpreta corretamente um cenário de falha, a precisão diminui com o aumento de falsos alarmes.

$$precisão = \frac{VP}{VP + FP}. \tag{4.1}$$

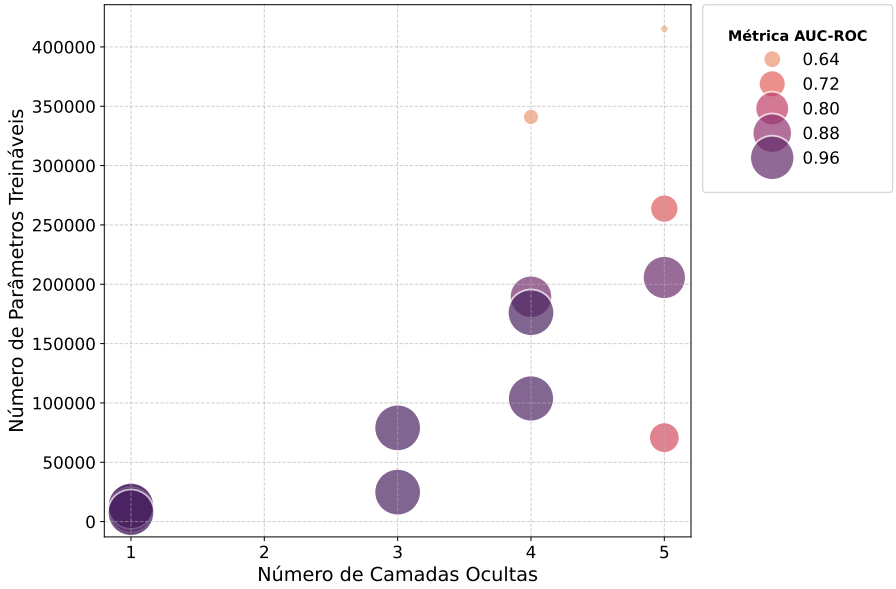
Recall

A métrica *recall* (ou sensibilidade) descrita na Equação 4.2 revela a taxa de VP entre todas as classes positivas do conjunto de dados. Ou seja, avalia quantas das amostras anômalas são classificadas corretamente. Considerando que a *recall* varia entre 0 e 1, quanto menor seu valor, maior o número de amostras anômalas sendo classificadas como normais.

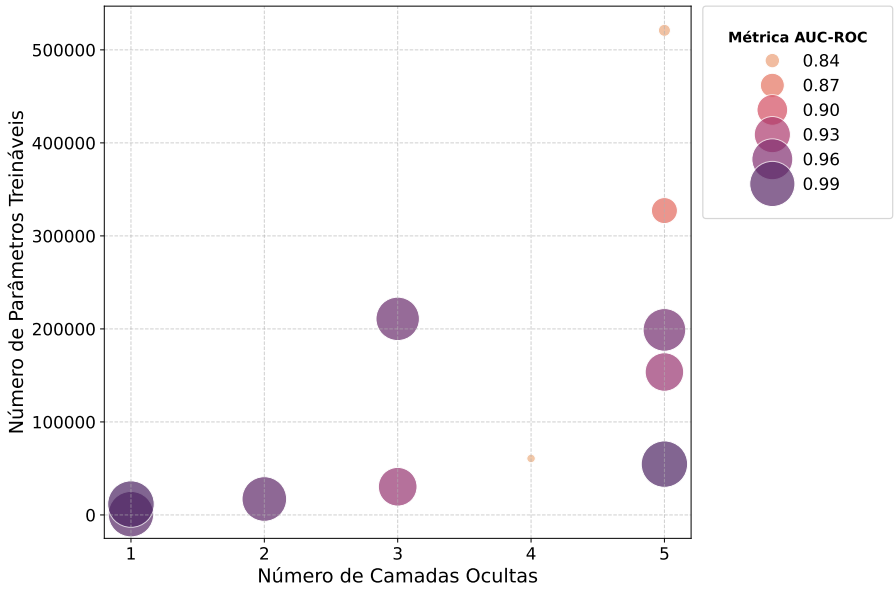
$$recall = \frac{VP}{VP + FN}. \tag{4.2}$$

AUC-ROC

A primeira métrica de avaliação utilizada para medir a qualidade do modelo é a AUC-ROC (*Area Under the Receiver Operating Characteristic Curve*). A curva ROC descreve a relação entre a taxa de verdadeiros positivos e a taxa de falsos positivos. A taxa de verdadeiros positivos é a relação entre os verdadeiros positivos e as amostras totais positivas do conjunto de dados. A taxa de falsos negativos é a



(a) Redes neurais selecionadas para o conjunto de dados MIMII.



(b) Redes neurais selecionadas para o conjunto de dados ToyADMOS.

Figura 4.3: Número de parâmetros treináveis das redes neurais selecionadas através da otimização Bayesiana.

relação entre os falsos negativos e todas as amostras positivas. A métrica AUC-ROC tem seu valor entre 0 e 1, sendo que o quanto mais próximo de 1, mais bem realizada é a tarefa de classificação.

F1-Score

A métrica *F1-Score* combina as métrias de precisão e *recall* e precisão conforme descrito na Equação 4.3. Assim, busca-se maximizar o valor desta métrica, alcançando

o valor 1, assim como a precisão e a *recall*.

$$F1 = \frac{2 \cdot recall \cdot precisão}{precisão + recall}. \tag{4.3}$$

MCC

A última métrica de avaliação adotada é o Coeficiente de correlação de Matthew (*Matthews Correlation Coefficient* – MCC). O MCC pode assumir valores entre -1 e 1, em que 1 indica uma predição perfeita, -1 uma predição completamente errada e 0 uma predição praticamente aleatória. Essa métrica mede a qualidade das classificações binárias e pode ser considerada mais confiável que outras métricas, como acurácia e Kappa de Cohen CHICCO *et al.* (2021). A métrica MCC é adequada para conjuntos de dados desbalanceados, pois sua formulação considera as predições corretas tanto nos casos positivos quanto nos negativos, não importando suas proporções no conjunto de dados original CHICCO e JURMAN (2020). O MCC é calculado da seguinte forma:

$$MCC = \frac{VP \times VN - FP \times FN}{\sqrt{(VP + FP) \times (VP + FN) \times (VN + FP) \times (VN + FN)}}. \tag{4.4}$$

4.2 Classificadores binários

Os classificadores binários são utilizados como base para a construção do modelo multiclasse e, posteriormente, a implementação do aprendizado federado. Dessa forma, os experimentos são divididos em duas fases de investigação. Na primeira, os algoritmos previamente apresentados são utilizados considerando sua relevância em trabalhos relacionados. Para corroborar ou contestar essa escolha, é utilizada uma biblioteca para automatizar e indicar os algoritmos mais promissores na segunda fase experimental. Logo, é considerada uma lista ainda mais extensa do que a adotada na primeira fase experimental.

4.2.1 Primeira fase experimental

Experimentos

Para primeira fase da etapa experimental, é utilizada uma máquina com CPU Intel Core i7-4790 e 16 GB de RAM. A biblioteca Librosa é adotada para extrair as características das amostras sonoras MCFEE *et al.* (2015). A biblioteca Scikit-learn PEDREGOSA *et al.* (2011) é adotada para a implementação dos algoritmos, com exceção das redes neurais e da etapa de balanceamento dos dados de treino. Para as redes neurais, a biblioteca utilizada é a Keras. Já a etapa de balanceamento

de dados através do SMOTE é realizada através da Imbalanced-learn[1]. Para o conjunto de dados MIMII, são utilizadas apenas as amostras com SNR = 6 dB. Esta escolha se deve ao fato de esta SNR ser a com maior facilidade de classificação.

Parâmetros e hiperparâmetros

Os hiperparâmetros adotados nos classificadores combinados no Super Learner são apresentados na Tabela 4.6. As duas redes neurais selecionadas e seus respectivos hiperparâmetros são apresentados nas Tabela 4.7 e 4.8 para os conjuntos de dados MIMII e ToyADMOS, respectivamente.

Tabela 4.6: Valores dos hiperparâmetros dos classificadores base.

Algoritmo	**Hiperparâmetro**	**Valor**
AB	`n_estimators`	50
KNN	`n_neighbors`	5
SVM	`C`	1
	`kernel`	rbf
	`gamma`	scale
RF	`n_estimators`	100
	`criterion`	Gini
	`max_features`	sqrt
	`min_samples_split`	2
	`min_samples_leaf`	1

Tabela 4.7: Hiperparâmetros das redes neurais adotadas para o conjunto de dados MIMII.

Hiperparâmetros	**Rede 1**	**Rede 2**
Nós por camada oculta	(320, 384, 256, 288, 384)	(288)
Dropout por camada	(0,0, 0,4, 0.0, 0,3, 0,3)	(0,4)
Taxa de aprendizado	0,001	0,01
Tipo das camadas	Densas	Densas
Função de Custo	Entropia Cruzada	Entropia Cruzada
Otimizador	Adam	Adam
Total de parâmetros treináveis	415.201	7.489

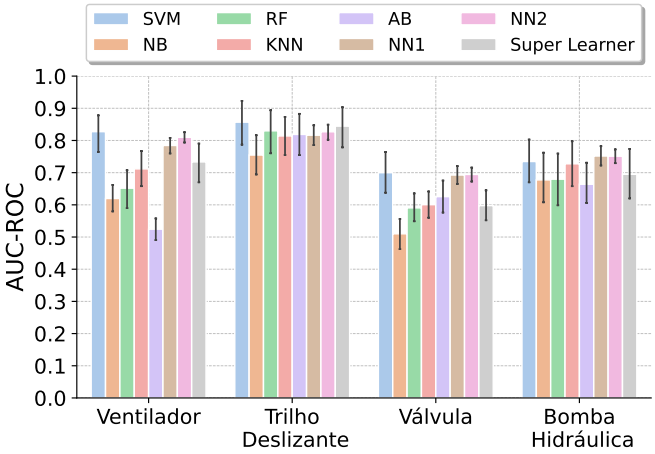
Resultados

A Figura 4.4 apresenta os resultados para a métrica AUC-ROC obtidos por todos os algoritmos, pelas redes neurais e pelo método Super Learner. Para o conjunto de dados MIMII, as redes neurais se destacam em comparação com os demais, conforme mostra a Figura 4.4a através das barras marrons e rosas.

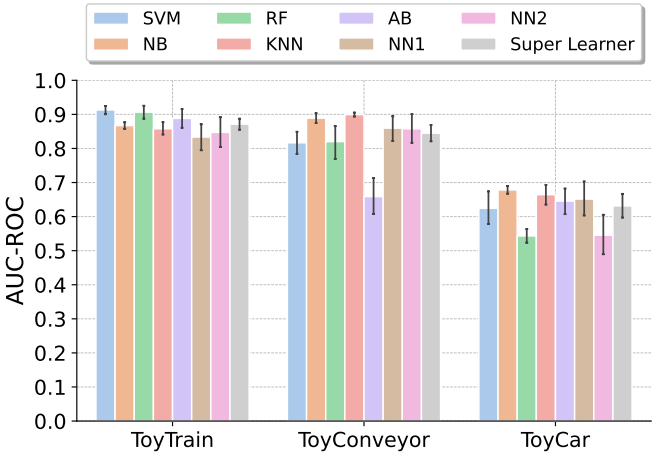
[1]Acessada em `https://imbalanced-learn.org/stable/references/generated/imblearn.over_sampling.SMOTE.html`.

Tabela 4.8: Hiperparâmetros das redes neurais adotadas para o conjunto de dados ToyADMOS.

Hiperparâmetros	**Rede 1**	**Rede 2**
Nós por camada oculta	(32, 352, 512, 352, 416)	(32)
Dropout por camada	(0,2, 0,4, 0,0, 0,3, 0,3)	(0,4)
Taxa de aprendizado	0,01	0,01
Tipo das camadas	Densas	Densas
Função de Custo	Entropia Cruzada	Entropia Cruzada
Otimizador	Adam	Adam
Total de parâmetros treináveis	520.993	833



(a) AUC-ROC para o conjunto de dados MIMII.

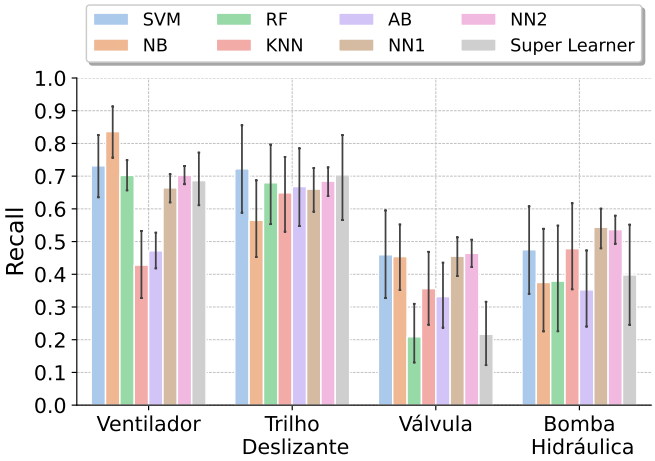


(b) AUC-ROC para o conjunto de dados ToyADMOS.

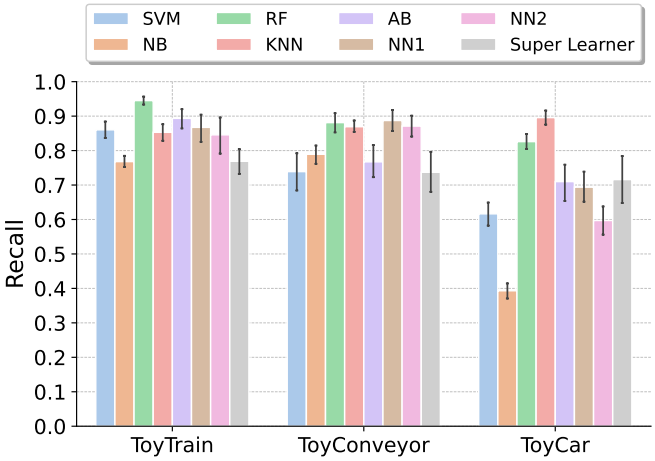
Figura 4.4: AUC-ROC para o aprendizado de máquinas tradicional.

Apesar do método Super Learner não obter melhores resultados que os demais algoritmos e se manter inferior ao SVM, a combinação é igual ou superior aos resul-

tados dos demais algoritmos utilizados como algoritmos base. Para o conjunto de dados ToyADMOS, os resultados são apresentados na Figura 4.4b. Nesse cenário, o algoritmo SVM não se destaca como para o outro conjunto de dados. Assim, não é possível apenas com essa métrica identificar o melhor algoritmo para a detecção de amostras anômalas.



(a) *Recall* para o conjunto de dados MIMII.

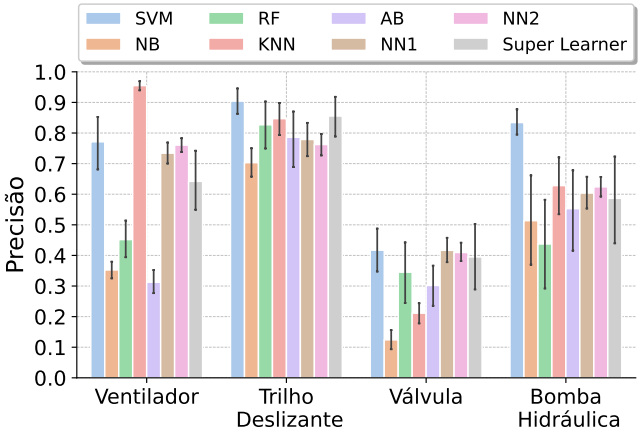


(b) *Recall* para o conjunto de dados ToyADMOS.

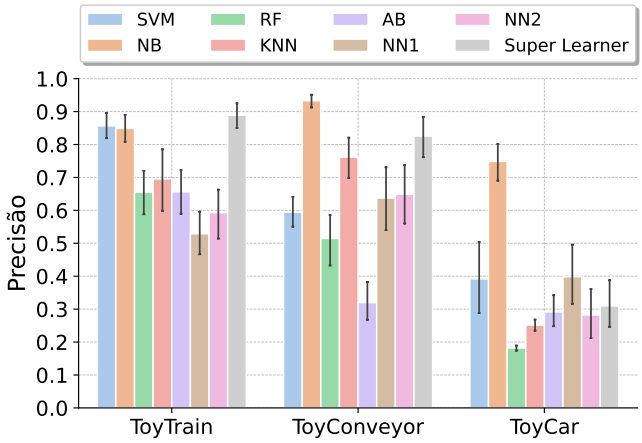
Figura 4.5: *Recall* para o aprendizado de máquinas tradicional.

A Figura 4.5a evidencia a alta taxa de falsos negativos no conjunto de dados MIMII, principalmente para o equipamento válvulas. Nesse cenário, cada equipamento possui um algoritmo que alcança melhor desempenho na classificação. Para os ventiladores, tem-se o algoritmo NB; para os trilhos deslizantes, os algoritmos SVM, as redes neurais e o método Super Learner; para as válvulas, SVM e NB; para as bombas, SVM, KNN e a rede neural 2. Analisando os resultados da métrica para o conjunto de dados ToyADMOS na Figura 4.5b, os algoritmos RF e KNN se destacam. Dessa forma, não é possível determinar um algoritmo que seja a melhor

escolha em todos os cenários.



(a) Precisão para o conjunto de dados MIMII.

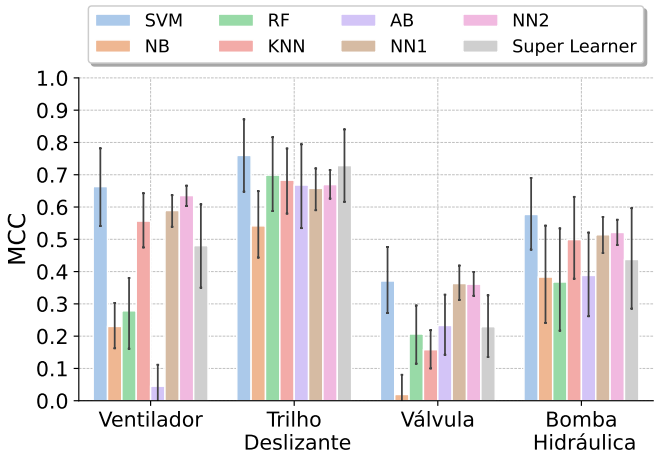


(b) Precisão para o conjunto de dados ToyADMOS.

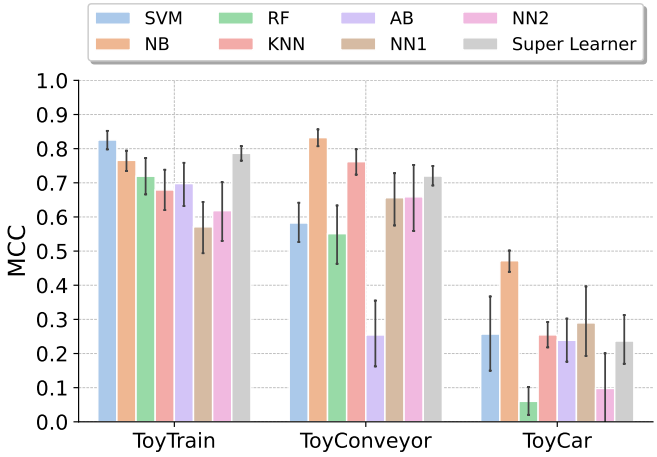
Figura 4.6: Precisão para o aprendizado de máquinas tradicional.

A métrica precisão que evidencia os alarmes falsos confirma a dificuldade na classificação do equipamento válvulas na Figura 4.6a. A métrica, em conjunto com as anteriores, demonstra que um algoritmo pode ser destaque ao observar determinado equipamento ou métrica, mas não ser em outros cenários. Apesar da proposta Super Learner apresentar resultados mais robustos, uma vez que os algoritmos encontram dificuldades em classificar corretamente, este método também apresenta resultados inferiores aos obtidos pelas redes neurais e o algoritmo SVM para o conjunto de dados MIMII, por exemplo. Por outro lado, para o conjunto de dados ToyADMOS, visto na Figura 4.6b, o algoritmo NB se sobressai em relação aos demais, enquanto no conjunto MIMII é um dos com piores resultados da métrica.

Por último, a métrica MCC fornece resultados mais adequados quanto a qualidade dos modelos. Através dessa métrica conclui-se que determinados algoritmos são



(a) MCC para o conjunto de dados MIMII.



(b) MCC para o conjunto de dados ToyADMOS.

Figura 4.7: MCC para o aprendizado de máquinas tradicional.

extremamente ineficazes para a classificação de anomalias em determinados equipamentos. Os algoritmos SVM, as redes neurais e o algoritmo Super Learner se mantêm como os melhores para o conjunto de dados MIMII na Figura 4.7a. Em equipamentos como ventiladores, os resultados do pior e melhor algoritmo saltam de 0.05 para por volta de 0.65 e, para as válvulas, saltam de próximo a 0 para por volta de 0.37. Os resultados ainda baixos comprovam também a dificuldade dos classificadores de estabelecer um modelo para a tarefa. Os resultados para o conjunto de dados ToyADMOS apresentado na Figura 4.7b mostra resultados um pouco superiores no geral, em que o método Super Learner e os algoritmos SVM e NB destacam-se em relação aos demais. Novamente, os resultados permitem concluir que, do ponto de vista das máquinas individuais, cada algoritmo torna-se o mais adequado dependendo da métrica a ser analisada. Esses resultados impulsionam a investigação de escolha de modelo utilizando o ponto de vista do cliente no aprendizado federado.

4.2.2 Segunda fase experimental

A segunda fase experimental segue com o objetivo de selecionar os melhores algoritmos para construir o modelo multiclasse a partir dos resultados obtidos pelos classificadores binários. Para isso, a biblioteca PyCaret é adotada para automatizar a busca. A biblioteca também contém os algoritmos utilizados na primeira fase experimental. O objetivo é manter ou excluir os algoritmos na construção do modelo multiclasse.

Seleção automatizada de algoritmos

Para a construção do sistema de detecção de defeitos multiclasse os algoritmos supervisionados são selecionados utilizando como base a biblioteca PyCaret e a classificação binária. Para isso, o tutorial obtido no repositório oficial é seguido[2]. O objetivo é, mais uma vez, buscar os algoritmos mais adequados de acordo com os conjuntos de dados disponíveis.

A classificação binária é utilizada como ponto inicial, seguindo a mesma abordagem mencionada anteriormente. O algoritmo SVM é adotado utilizando os mesmos parâmetros do trabalho anterior GANTERT *et al.* (2021) e os demais parâmetros são selecionados através do suporte da biblioteca PyCaret.

A métrica AUC-ROC é selecionada para escolha dos algoritmos supervisionados, uma vez que a métrica é usada para avaliar o modelo base proposto pelos autores do conjunto de dados MIMII. Os quatro melhores algoritmos nessa etapa são selecionados considerando o número de vezes que cada um apresentou o melhor desempenho nos experimentos. A biblioteca realiza a comparação com diversos algoritmos, incluindo os utilizados em resultados prévios, exceto pelo SVM. Dessa forma, o algoritmo SVM é adicionado à lista por seu desempenho anterior. Os novos cinco algoritmos supervisionados selecionados são, então:

- **Extremely Randomized Trees Classifier (ERT):** O ERT é um método *ensemble* em que árvores de decisão aleatórias são treinadas e a classificação é realizada através da maioria de votos.
- **Linear Discriminant Analysis (LDA):** O algoritmo LDA cria um hiperplano para separar as classes adotando as características das amostras como dimensões. O hiperplano é criado para maximizar a distância entre as médias das classes e minimizar a variação entre as classes.
- **Gradient Boosting (GB):** O Gradient Boosting é um método *boosting ensemble*, ou seja, os classificadores são treinados sequencialmente através da

[2]Accessed at `https://github.com/pycaret/pycaret/blob/master/tutorials/Tutorial-BinaryClassification.ipynb`.

informação dos classificadores anteriores para adaptar e melhorar o modelo final. Nesse algoritmo, o objetivo é minimizar a função de perda utilizando o gradiente para atualizar os pesos de cada classificador.

- **Light Gradient Boosting Machine (LGBM):** O LGBM difere do GB ao reduzir o número de amostras, utilizando uma técnica de amostragem denominada *Gradient-based One-Side Sampling*. Além disso, adota o *Exclusive Feature Bundling* para reduzir a dimensionalidade das características. Assim, a complexidade do algoritmo e o consumo de memória são reduzidos, tornando o treinamento dos modelos mais rápido.

Dessa forma, os algoritmos supervisionados podem ser divididos em dois tipos: os baseados em hiperplanos e os baseados em algoritmos *ensemble*. O LDA e o SVM pertencem ao primeiro tipo, e o ERT, o GB e o LGBM são baseados em métodos *ensemble*. Esses três últimos algoritmos adotam árvores de decisão como classificadores.

Resultados

Para os próximos experimentos da classificação binária e multiclasse é utilizado o Google Colab com CPU Intel(R) Xeon(R) 2.20GHz e 12GB de RAM. São listados os seguintes resultados da segunda fase experimental:

(**a**) os experimentosda primeira fase experimental são replicados usando uma ferramenta automatizada para selecionar os algoritmos com melhor desempenho. A ideia é observar se são possíveis melhorias de desempenho na classificação binária.

(**b**) as propostas multiclasse são implementadas misturando parcial e totalmente as relação sinal-ruído para construir um modelo mais genérico e leve para identificar o status de equipamentos industriais com diferentes níveis de ruído de fundo. Em seguida, a proposta é comparada à abordagem binária, considerando seu desempenho de classificação. Por fim, as matrizes de confusão das abordagens multiclasse propostas usando os algoritmos com melhor desempenho nos resultados anteriores são apresentadas e analisadas;

(**c**) o tempo de treinamento do modelo e o poder de generalização dos modelos multiclasse parcial e totalmente mistos são comparados com os resultados obtidos pelos modelos binários.

A Tabela 4.9 apresenta os hiperparâmetros mais relevantes utilizados pelos algoritmos implementados. Esses hiperparâmetros são os valores padrão definidos pela

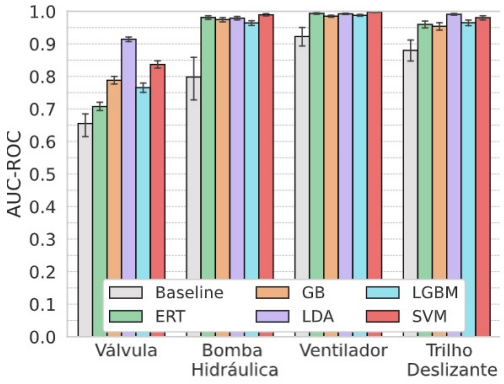
Tabela 4.9: Hiperparâmetros dos algoritmos de aprendizado de máquinas selecionados.

Algoritmo	**Hiperparâmetro**	**Valor**
ERT	`criterion`	Gini
	`n_estimators`	100
LDA	`solver`	svd
LGBM	`n_estimators`	100
	`learning_rate`	0.1
	`boosting_type`	gbdt
GB	`criterion`	Friedman_mse
	`n_estimators`	100
	`learning_rate`	0.1

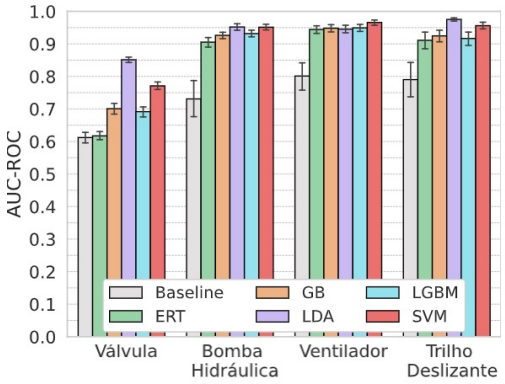
biblioteca PyCaret, uma vez que a biblioteca é utilizada para a seleção de modelos. No algoritmo ERT, a função de impureza de Gini é adotada e 100 estimadores utilizados; no LDA a técnica de Decomposição em Valores Singulares (*Singular Value Decomposition* – SVD) é selecionada, sendo particularmente recomendada em casos com alto número de características. Em LGBM, o número de estimadores é igual a 100 e a taxa de aprendizado é igual a 0,1 e o tipo de *boosting* selecionado é o *Traditional Gradient Boosting Decision Tree*. No GB, o critério adotado é o *Friedman's Mean Squared Error*.

As métricas AUC-ROC e F1-Score são analisadas considerando o conjunto de dados MIMII e os trabalhos relacionados GANTERT *et al.* (2021, 2022). Para isso, os resultados são divididos de acordo com o tipo de equipamento, em que as barras representam os diferentes IDs dos equipamentos. Considerando os resultados publicados pelos autores do conjunto de dados MIMII, representados pelas barras cinzas como o trabalho de base que utiliza como modelo um autoencoder com cinco camadas: 8 neurônios na camada de gargalo e 64 neurônios nas demais. Os resultados apresentados adotam um intervalo de confiança de 95% . As Figuras 4.8a,4.8c apresentam a AUC-ROC para as três diferentes SNRs, 6 dB, 0 dB e −6 dB, respectivamente.

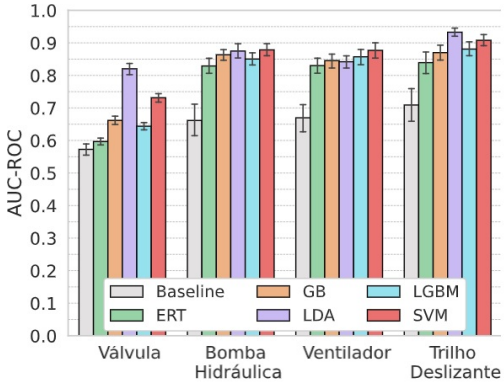
Para as válvulas, tanto o trabalho utilizado como base de comparação quanto ERT (representado pelas barras verdes) alcançam resultados similares, entre 0.55 and 0.6 para as diferentes SNR indicando baixa qualidade de classificação. Entretanto, LDA e LR, as barras roxas e vermelhas, respectivamente, alcançam resultados de AUC-ROC superiores ou próximos a 0,8 para todos os SNRs. Esses resultados indicam que os algoritmos baseados em hiperplanos (LDA e SVM), além de melhorarem o desempenho do modelo baseado em redes neurais, atingem valores mais altos para a métrica AUC-ROC em comparação com algoritmos baseados em *ensemble*. Ao comparar o desempenho dos algoritmos selecionados segundo a métrica F1-Score, observamos que os algoritmos baseados em hiperplanos se destacam tanto para válvulas quanto para trilhos deslizantes em todos os SNRs, conforme mostrado



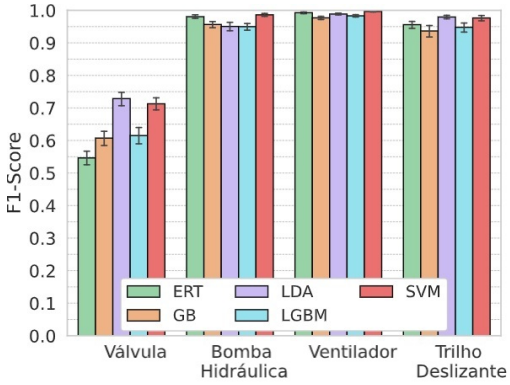
(a) AUC-ROC para SNR = 6 dB.



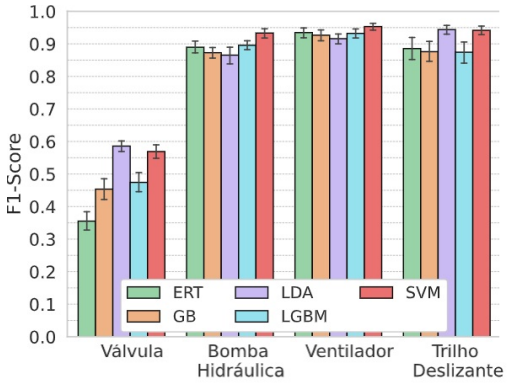
(b) AUC-ROC para SNR = 0 dB.



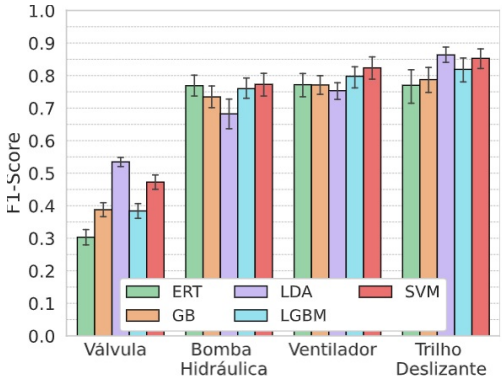
(c) AUC-ROC para SNR = −6 dB.



(d) F1-Score para SNR = 6 dB.



(e) F1-Score para SNR = 0 dB.

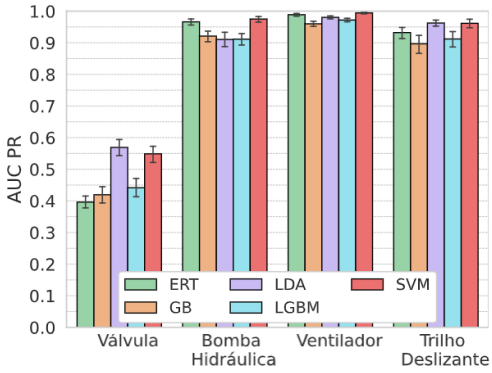


(f) F1-Score para SNR = −6 dB.

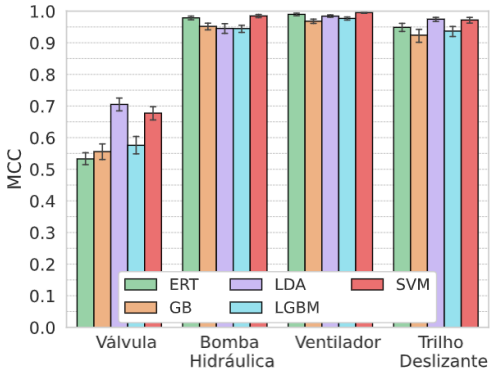
Figura 4.8: AUC-ROC e F1-Score para os diferentes algoritmos e as distintas SNR.

nas Figuras 4.8d,4.8f. Os experimentos para a métrica F1-Score não contêm resultados do trabalho de base, pois os autores apenas fornecem resultados para a métrica AUC-ROC.

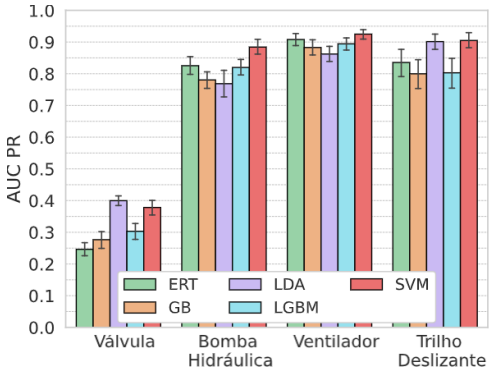
A Figura 4.9 reforça os resultados obtidos para as métricas AUC-ROC e F1-Score através das métricas AUC-PR e MCC. O LDA e o SVM apresentam resultados superiores para válvulas e trilhos deslizantes, em comparação com métodos *ensemble*, em todas as SNRs. Considerando apenas os algoritmos baseados em *ensemble*, o LGBM representado pelas barras azuis alcançam os maiores valores de métricas em comparação com o ERT e o GB para a relação SNR mais baixa e, portanto, a mais



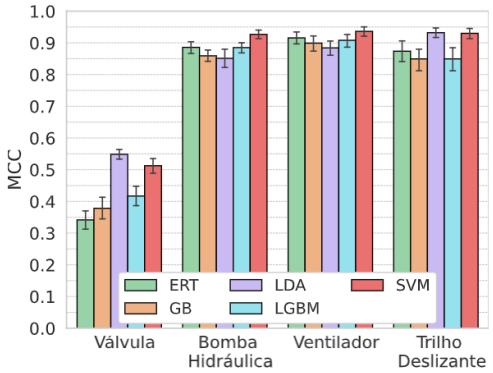
(a) AUC-PR para SNR = 6 dB.



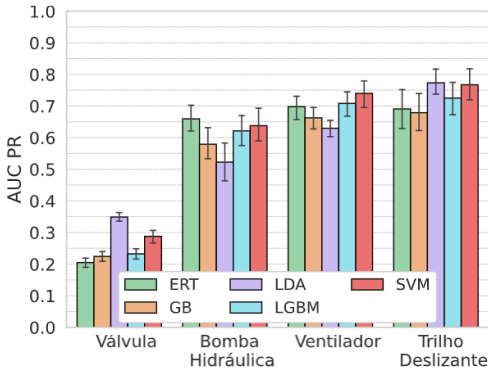
(b) MCC com SNR = 6 dB.



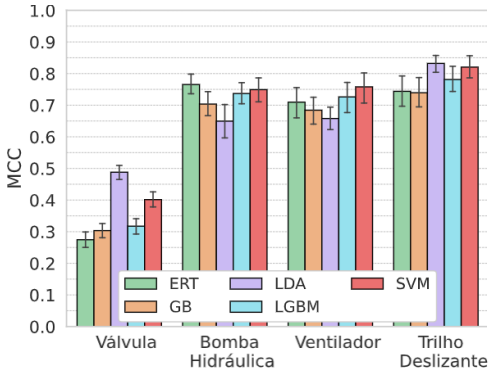
(c) AUC-PR para SNR = 0 dB.



(d) MCC para SNR = 0 dB.



(e) AUC-PR para SNR = −6 dB.



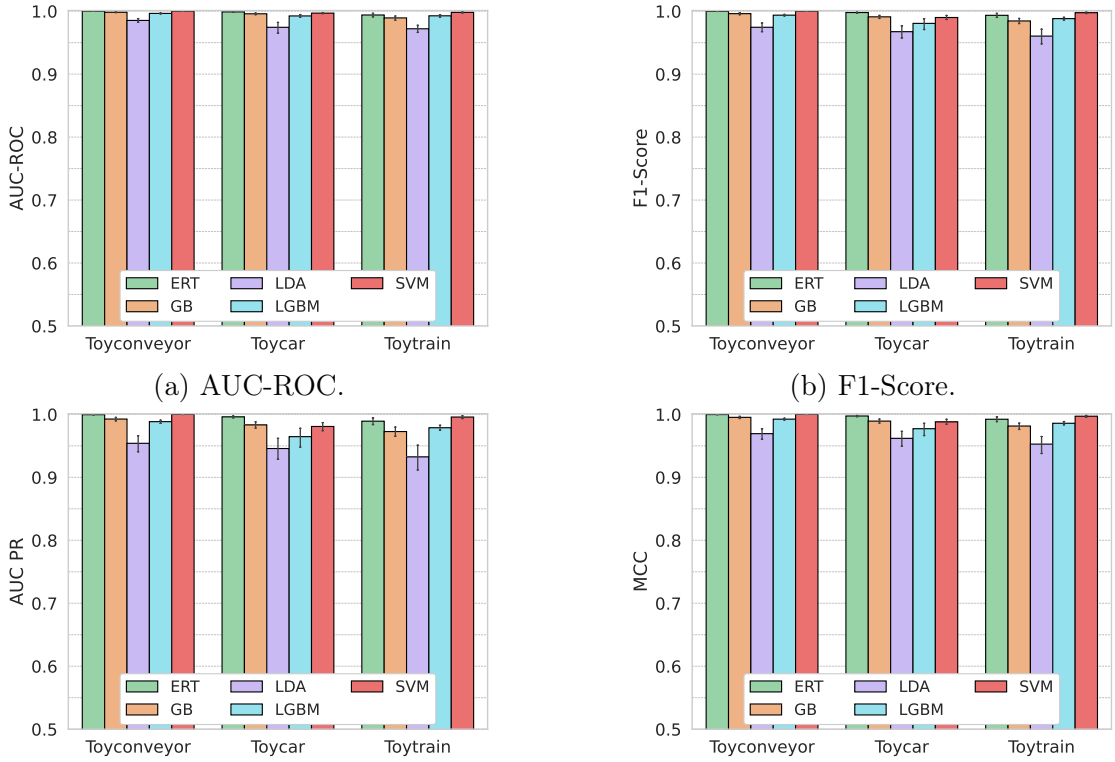
(f) MCC para SNR = −6 dB.

Figura 4.9: AUC-PR e MCC para os diferentes algoritmos e as distintas SNR.

desafiadora conforme apresentam as Figuras 4.8c, 4.8f, 4.9e e 4.9f. Dessa forma, os algoritmos SVM e LGBM são selecionados para representar ambos os tipos de algoritmos na propostas multiclasse: baseados em hiperplanos e em *ensemble*.

Os resultados utilizando o conjunto de dados ToyADMOS validam as propostas multiclasse analisadas com o conjunto de dados MIMII. Assim, os passos adotados após a etapa de seleção dos modelos binários com a biblioteca PyCaret são replicados.

A Figura 4.10 apresenta os resultados obtidos para a classificação binária. Os algoritmos selecionados nas etapas anteriores também são adotados neste experi-



(a) AUC-ROC.

(b) F1-Score.

(c) AUC PR.

(d) MCC.

Figura 4.10: AUC-ROC, F1-Score, AUC PR e MCC para o conjunto de dados ToyADMOS.

mento. O desempenho observado nesse conjunto de dados é superior ao obtido para o conjunto de dados MIMII, alcançando valores acima de $0,95$ para métrica AUC-ROC para todos os algoritmos e equipamentos analisados, conforme apresenta a Figura 4.10a. O padrão é repetido para as métricas F1-Score, AUC-PR e MCC, nas Figuras 4.10b, 4.10c e 4.10d, respectivamente. Esses resultados mostram que as características espectrais e os algoritmos de aprendizado de máquina são efetivos para a detecção de eventos sonoros, mesmo utilizando outras fontes de áudio como entrada.

4.3 Classificadores multiclasse

A classificação multiclasse considera que ter um modelo personalizado para cada configuração específica não é prático em cenários reais. Por isso, a abordagem multiclasse busca reduzir o número de modelos combinando classificadores binários. Essa estratégia reduz a quantidade de classificadores e modelos ao agrupar amostras, desconsiderando certas propriedades particulares. Assim, em vez de utilizar diferentes modelos binários para inferir se um equipamento está de acordo com o padrão de funcionamento esperado, são empregados menos modelos ou um único modelo multiclasse capaz de fornecer a mesma informação.

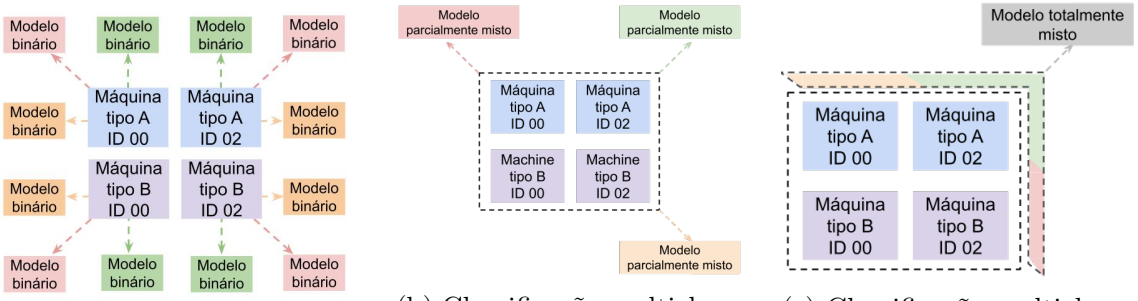
A Figura 4.11 ilustra essa ideia utilizando o conjunto de dados MIMII. A Figura 4.11a apresenta a classificação binária com dois tipos de equipamentos (Máquina Tipo A e B) e duas máquinas individuais de cada tipo (ID 00 e 02), onde as setas coloridas indicam diferentes níveis de SNR. Cada máquina individual requer um modelo binário para cada nível de SNR. O resultado de cada modelo é simplesmente uma indicação de se o equipamento está funcionando corretamente. No total, o número de modelos binários é igual ao número de tipos de máquina $\times$ número de máquinas $\times$ número de níveis de SNR.

A Figura 4.11b mostra a ideia por trás da classificação multiclasse parcialmente mista. Os equipamentos compartilham o mesmo modelo, mas ainda há separação por nível de SNR. Portanto, os modelos utilizados devem ser multiclasse, pois precisam indicar tanto o tipo de equipamento quanto seu estado operacional. Nesse exemplo, o número de modelos é igual ao número de níveis de SNR, e o número de classes corresponde ao número de tipos de equipamento multiplicado pelos estados possíveis. Já a Figura 4.11c apresenta o modelo multiclasse totalmente misto, onde um único modelo é treinado com amostras de todos os níveis de SNR. Diferentemente da proposta anterior, o número de modelos é unitário, pois todas as amostras são agrupadas independentemente do nível de SNR.

A saída dos modelos multiclasse indica o tipo de equipamento e seu estado. Contudo, como visto na abordagem totalmente mista, é possível reduzir o número de modelos para apenas um ao eliminar a necessidade de conhecer previamente o nível de SNR da amostra de áudio recebida. Essa abordagem aumenta a generalidade da proposta, reduzindo o número de modelos a serem treinados. A motivação baseia-se na constatação de que, em uma planta industrial, o nível de SNR de uma amostra de áudio provavelmente é desconhecido.

Além disso, saber que um equipamento de determinado tipo não está funcionando corretamente pode ser suficiente, em contraste com um modelo mais sofisticado que identificaria a máquina específica com falha. Ao comparar as três abordagens, partimos de um número total de modelos igual ao número de tipos de máquina $\times$ número de máquinas $\times$ número de níveis de SNR, o que resulta em 48 modelos no caso do conjunto de dados MIMII. Na abordagem parcialmente mista, o número de modelos passa a ser igual ao número de níveis de SNR, ou seja, 3. Considerando que cada modelo possui 8 classificadores, o total de classificadores utilizados é 24. Por fim, a abordagem totalmente mista reduz para um único modelo com 8 classificadores.

Essa mudança é fundamental em ambientes industriais, onde não se pode antecipar qual equipamento gerou o som recebido. Esse conhecimento seria necessário para selecionar o modelo binário correspondente. Com a abordagem multiclasse, é possível utilizar um modelo independente do tipo de equipamento gerador. Essas alternativas conferem maior generalidade à proposta, o que é desejável em cenários



(a) Classificação Binária.

(b) Classificação multiclasse parcialmente mista.

(c) Classificação multiclasse totalmente mista.

Figura 4.11: Classificação binária e modelos multiclasse propostos em busca da generalização da classificação em cenários industriais.

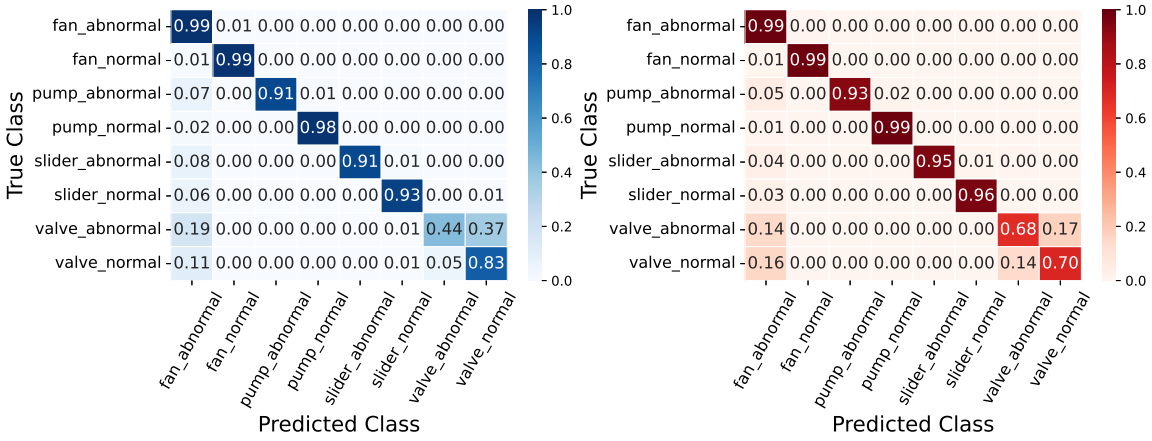
reais. Vale destacar que as características como tipo de equipamento e nível de SNR são mencionadas apenas como exemplo com base no conjunto de dados MIMII. A mesma ideia pode ser aplicada a outros conjuntos de dados, como o ToyADMOS.

A classificação multiclasse totalmente mista aprimora ainda mais a generalidade da proposta ao não distinguir entre diferentes níveis de SNR. A mesma abordagem é aplicada ao conjunto de dados ToyADMOS, onde são consideradas apenas a classificação binária e a multiclasse totalmente mista. Nesse caso, reduzimos o número de classificadores de 11 na classificação binária para 6 na abordagem multiclasse totalmente mista. Reduzimos o número de classificadores com o objetivo de simplificar o problema de classificação. Assim, o objetivo é substituir modelos binários específicos por modelos multiclasse genéricos.

A abordagem multiclasse adota a estratégia One-vs-All (também conhecida como One-vs-Rest), na qual o número de classificadores em cada modelo é igual ao número de classes. Cada classificador é ajustado contra todas as outras classes, de forma semelhante a um problema de classificação binária. Em outras palavras, a classificação multiclasse é dividida em múltiplos problemas binários. Os classificadores com maiores métricas determinam a predição final

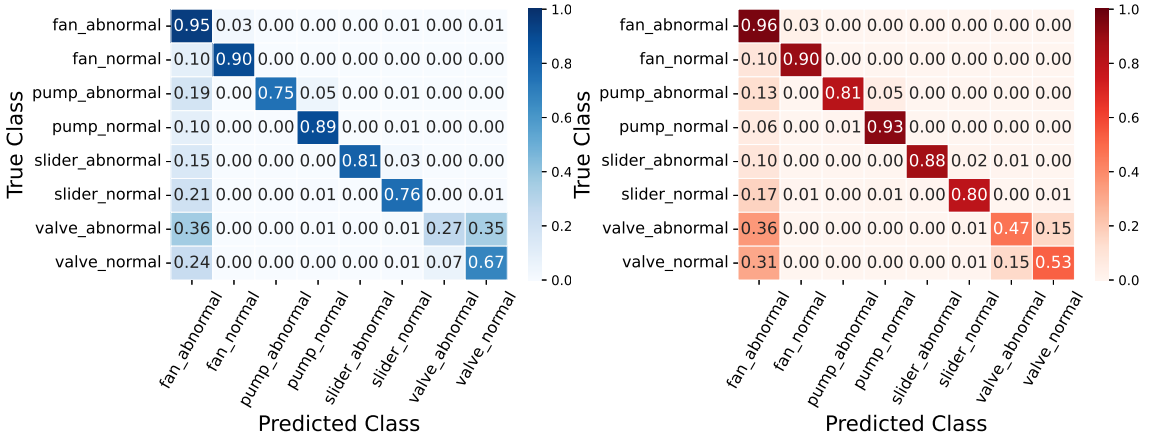
A motivação para a classificação multiclasse baseia-se na redução do número de modelos necessários. Para o conjunto de dados MIMII são necessários 48 classificadores binários, considerando as três SNRs disponíveis, os quatro tipos de equipamentos e as 4 máquinas individuais disponíveis para cada tipo de equipamento. Cada classificador tem como saída se a máquina individual encontra-se na condição normal ou anômala de operação. Por outro lado, as propostas multiclasse com mistura parcial de SNR necessitam apenas de 3 classificadores, sendo um para cada nível de SNR, e 8 classes distintas que indicam o tipo de equipamento e seu estado de operação simultaneamente.

A proposta totalmente misturada também é implementada para o algoritmo ToyADMOS. Assim como o conjunto de dados MIMII, as máquinas do mesmo tipo



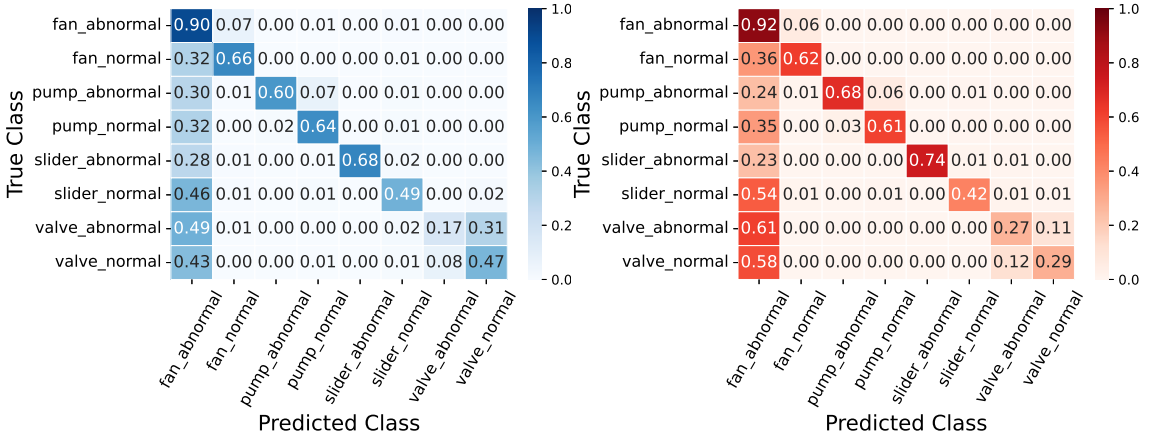
(a) LGBM para SNR = 6 dB.

(b) SVM para SNR = 6 dB.



(c) LGBM para SNR = 0 dB.

(d) SVM para SNR = 0 dB.

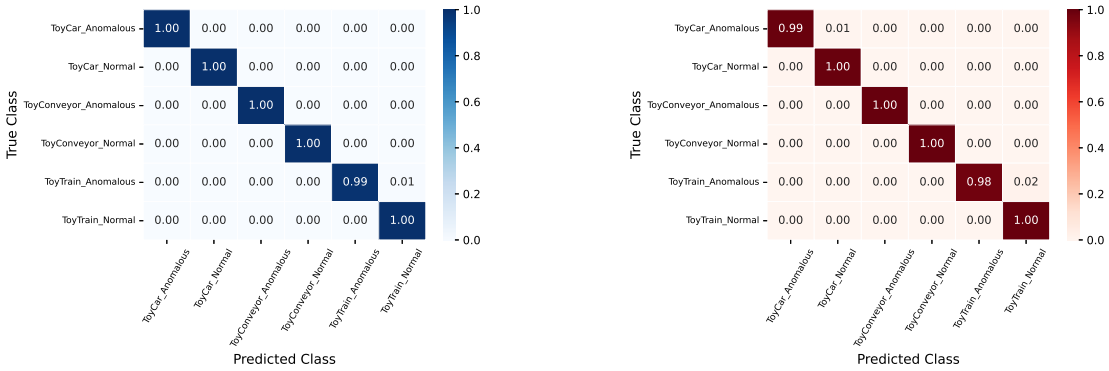


(e) LGBM para SNR = −6 dB.

(f) SVM para SNR = −6 dB.

Figura 4.12: Matrizes de confusão dos algoritmos LGBM e SVM para classificação multiclasse mistura parcial separadas pelas diferentes SNRs.

são agrupadas para criar uma classe de equipamento que contenha também a informação de seu respectivo estado. A Figura 4.13a apresenta a matriz de confusão para o algoritmo LGBM. O estado anômalo é classificado corretamente em 100% das vezes para os equipamentos ToyCar e ToyConveyor, e 99% para o equipamento ToyTrain. Para o algoritmo SVM, ilustrado na Figura 4.13b, a classificação dos estados anômalos é correta em 100% para as amostras do equipamento ToyConveyor 99% para o equipamento ToyCar e 98% para as amostras do equipamento ToyTrain. O desempenho superior dos modelos binários pode explicar o desempenho também superior da proposta totalmente mista para o conjunto de dados ToyADMOS. Da mesma forma, a principal razão para o melhor desempenho em comparação com o conjunto de dados MIMII é o maior número de amostras disponíveis para treinamento.

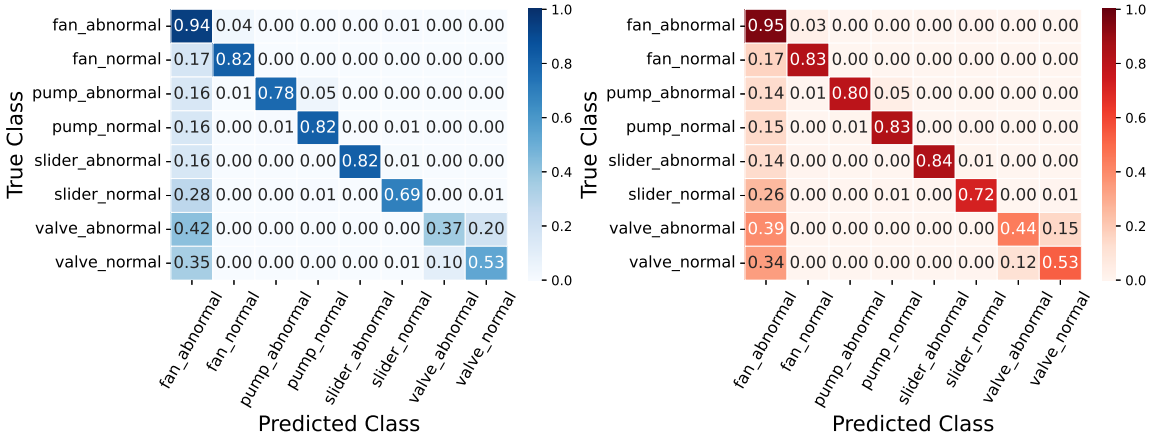


(a) LGBM. (b) SVM.

Figura 4.13: Matrizes de confusão para os algoritmos LGBM e SVM na classificação multiclasse para o conjunto de dados ToyADMOS.

Dessa forma, são necessários 24 classificadores. Ao utilizar mistura total SNR, é possível reduzir o número total de classificadores para 8, ou seja, 1/3 comparado a abordagem com mistura parcial de SNR. Assim, é possível reduzir o número de classificadores em até 83%. O conjunto de dados fornece a estrutura apropriada para a classificação binária e cada modelo é treinado usando os dados das máquinas individuais.

A avaliação das propostas multiclasse utiliza, inicialmente, apenas o conjunto de dados MIMII. A abordagem parcialmente mista mantém os níveis de SNR separados, enquanto a abordagem totalmente mista reúne amostras com diferentes níveis de SNR. A abordagem multiclasse totalmente mista visa reproduzir um cenário mais realista, em que a classificação não pode se basear em amostras separadas por níveis de SNR. Em seguida, a análise é dividida considerando o desempenho da classificação, a quantidade de recursos computacionais para treinamento e a capacidade de generalização de todos os modelos. Dessa forma. as abordagens multiclasse são mais generalistas. Portanto, o objetivo é propor modelos de uso geral que ainda possam



(a) LGBM com mistura completa de SNRs. (b) SVM com mistura completa de SNRs.

Figura 4.14: Matrizes de confusão para os algoritmos LGBM e SVM para a classificação multiclasse com SNRs totalmente misturadas utilizando todas as amostras.

ter um bom desempenho mesmo nos piores cenários.

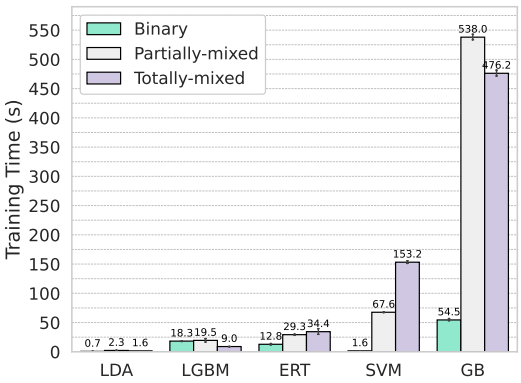
A Figura 4.12 apresenta as matrizes de confusão obtidas para os algoritmos LGBM e SVM na abordagem parcialmente misturada e correspondem às médias calculadas após 10 rodadas de experimentos. A escolha desses algoritmos se deve ao desempenho obtido em métricas globais de AUC-ROC e F1-Score. As Figuras 4.12a e 4.12b apresentam os resultados para os experimentos cuja SNR é igual a 6 dB. O LGBM obteve taxas de acerto de 99% na classificação da classe anormal para ventiladores, 88% para bombas e 91% para trilhos deslizantes. Já o SVM apresentou desempenho entre 94% e 99% na identificação de amostras anormais. Com exceção dos resultados referentes às válvulas, ambos os algoritmos demonstraram capacidade de identificar corretamente o tipo de equipamento e seu respectivo estado.

Na proposta multiclasse totalmente misturada, amostras com diferentes níveis de SNR (6 dB, 0 dB e −6 dB) são agrupadas. A estratégia possibilita a construção de um modelo mais genérico, aumenta o número de amostras por classe disponível para o treinamento e teste dos modelos e, ao mesmo tempo, procura reproduzir um cenário mais realista, onde sinais acústicos apresentam diferentes condições de ruído. As Figuras 4.14a e 4.14b apresentam as matrizes de confusão para o LGBM e o SVM, respectivamente. Assim como observado anteriormente, os resultados continuam com taxa de erro reduzida. O comportamento observado para os ventiladores exemplifica essa tendência: o LGBM classificou corretamente falhas em 82% dos casos, enquanto o SVM obteve 71%. Além disso, o LGBM alcança 82% de acerto na identificação do estado anormal em bombas. Assim, em relação ao modelo parcialmente misturado, houve uma redução aproximada de 10% no desempenho, compensada pela maior generalização e simplificação do modelo.

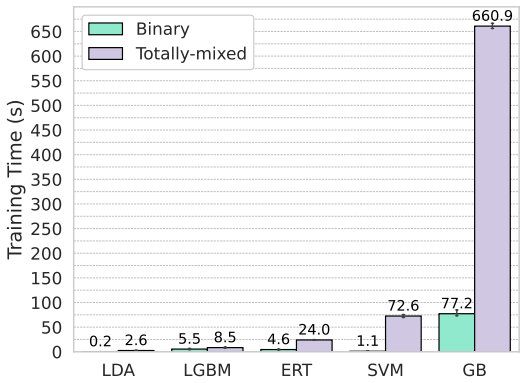
4.3.1 Recursos computacionais para treinamento do modelo

Os resultados a seguir avaliam os possíveis compromissos da abordagem generalizada. A Figura 4.15b apresenta o tempo de treinamento médio para os modelos considerando todas as rodadas executadas. As propostas multiclasse parcial e totalmente misturadas são comparadas com a abordagem binária. O tempo necessário é a soma de todos os modelos treinados em sequência utilizando a mesma configuração computacional.

É possível observar que o tempo de treinamento depende do algoritmo adotado. Para o LDA, o LGBM e o ERT é aproximadamente o mesmo para todas as propostas. Entretanto, para o SVM e o GB, o tempo aumenta consideravelmente nos modelos multiclasse. Isso evidencia que, dependendo do algoritmo, o número de amostras impacta de forma mais evidente o tempo necessário para treinamento do algoritmo. Vale ressaltar que o tempo de treinamento não afeta a extração das características de som do conjunto de dados original, uma vez que esta etapa é realizada offline.



(a) Tempo de treinamento para as duas abordagens utilizando o conjunto de dados MIMII.



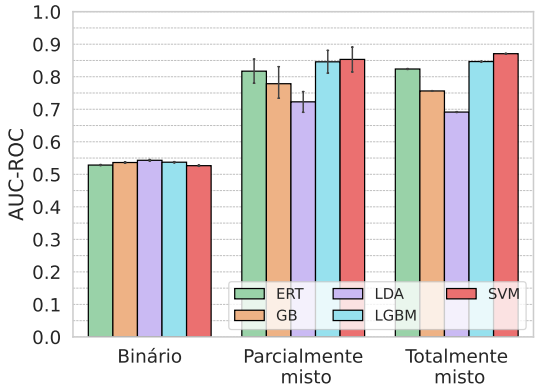
(b) Tempo de treinamento para as duas abordagens para o conjunto de dados ToyADMOS.

Figura 4.15: Comparação do tempo de treinamento para os classificadores binários e multiclasse.

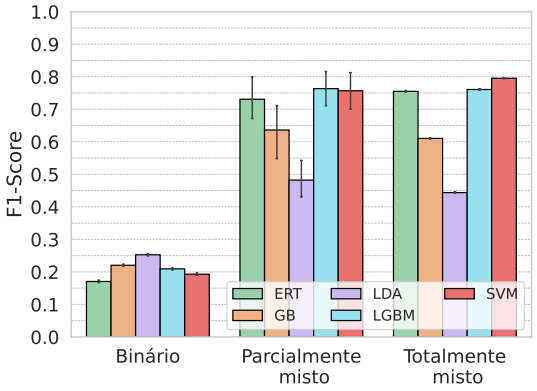
A Figura 4.15b apresenta o tempo de treinamento do modelo. Mais uma vez o tempo aumenta na proposta multiclasse, assim como no experimento utilizando o conjunto de dados MIMII.

4.3.2 Generalização do modelo

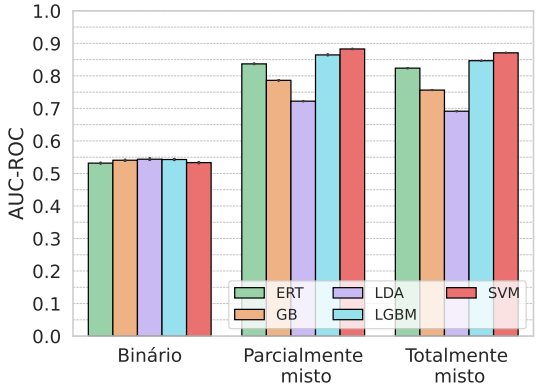
Esta etapa analisa a importância da generalização do modelo. Os modelos binários são testados de forma cruzada com amostras de som de equipamentos não utilizados no treinamento. Por exemplo, o modelo binário treinado para ventiladores tem seu desempenho testado com amostras de bombas hidráulicas. De forma similar, os modelos multiclasse com mistura parcial das amostras são testados com amostras de



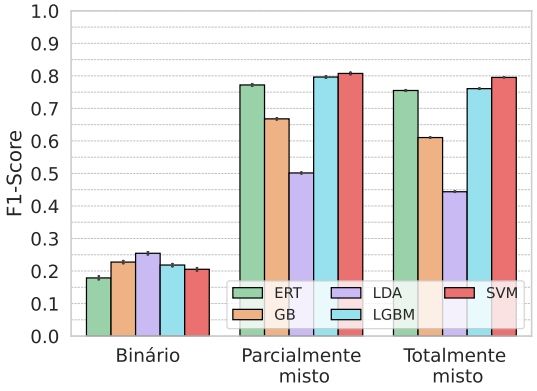
(a) AUC-ROC para SNR = 6 dB.



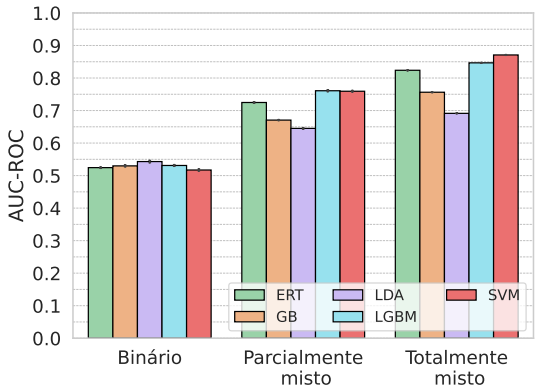
(d) F1-Score para SNR = 6 dB.



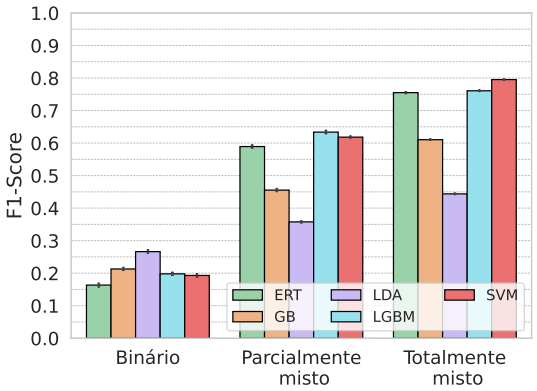
(b) AUC-ROC para SNR = 0 dB.



(e) F1-Score para SNR = 0 dB.



(c) AUC-ROC para SNR = −6 dB.

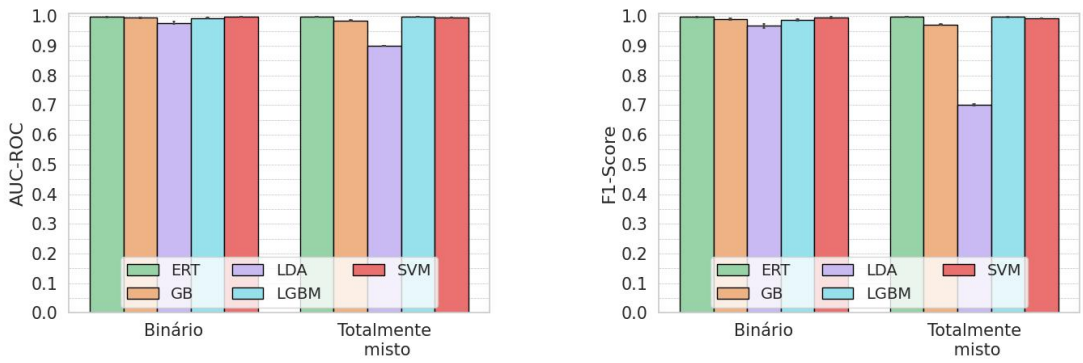


(f) F1-Score para SNR = −6 dB.

Figura 4.16: AUC-ROC e F1-Score nas três abordagens para o conjunto de dados MIMII.

diferentes níveis de SNR. Por exemplo, o modelo com mistura parcial treinado para SNR de 6 dB é testado com amostras de SNR igual a −6 dB. O modelo multiclasse único com mistura total de amostras não muda, uma vez que é treinado com todas as amostras do conjunto de dados.

Ao analisar a métrica AUC-ROC nas Figuras 4.16a, 4.16b e 4.16c, os modelos binários não excedem 0,65 em nenhum dos SNRs analisados. Por outro lado, os



(a) AUC-ROC. (b) F1-Score.

Figura 4.17: AUC-ROC e F1-Score para as duas abordagens para o conjunto de dados ToyADMOS.

resultados do modelo multiclasse revelam que, mesmo testados em condições mais ruidosas, os algoritmos LGBM e SVM alcançam AUC-ROC igual a aproximadamente $0,85$ em SNRs de $6\,\text{dB}$ e $0\,\text{dB}$. As Figuras 4.16a, 4.16b e 4.16c apresentam os resultados das abordagens considerando a métrica F1-Score. A baixa generalização do modelo binário se torna mais evidente através dos valores da métrica AUC-ROC abaixo de $0,3$ para todos os valores de SNR. Assim como nas métricas anteriores, os modelos multiclasse alcançam resultados superiores para níveis mais altos de SNR. No entanto, a abordagem multiclasse totalmente mista supera todas as outras alternativas em cenários com ruído considerável, atingindo AUC-ROC e F1-Score de aproximadamente $0,8$ com o algoritmo SVM. Dessa forma, o resultado indica uma tendência para pesquisas futuras sobre o poder dos modelos genéricos quando modelos específicos não se adequam bem à quantidade de ruído.

É importante destacar que não se espera que um modelo treinado para um determinado nível de SNR apresente bom desempenho quando aplicado a amostras com níveis distintos de SNR. Nesse sentido, a abordagem totalmente misturada demonstra que um modelo treinado considerando todos os níveis de SNR supera aquele ajustado especificamente para um nível particular, caso utilizado para classificar amostras sonoras obtidas em diferentes condições. Caso não seja possível estimar previamente o nível de SNR de uma amostra recebida, a seleção do modelo especializado adequado torna-se uma tarefa desafiadora. Nessa situação, a proposta multiclasse se mostra vantajosa, pois fornece simultaneamente o estado operacional (normal ou anormal) e o tipo de equipamento como resultado, independentemente do nível de SNR.

A avaliação cruzada para medir a generalização do modelo é apresentada nas Figuras 4.17a e 4.17b. As classificações binária e multiclasse são comparadas através das métricas AUC-ROC e F1-score, respectivamente. Conforme observado ante-

riormente, os classificadores binários alcançam valores relativamente inferiores nas métricas em comparação com a proposta multiclasse quando testados com amostras de máquinas distintas. Isso evidencia o quão específicos os modelos binários são. Para a métrica AUC-ROC, o valor é pouco superior a $0,6$, permanecendo com o F1-Score abaixo de $0,4$. Entretanto, o modelo totalmente misto atinge o valor máximo nos algoritmos ERT, LGBM e SVM, uma vez que esses modelos são treinados com todos os dados disponíveis.

Capítulo 5

Seleção de modelos baseada em avaliação das federações

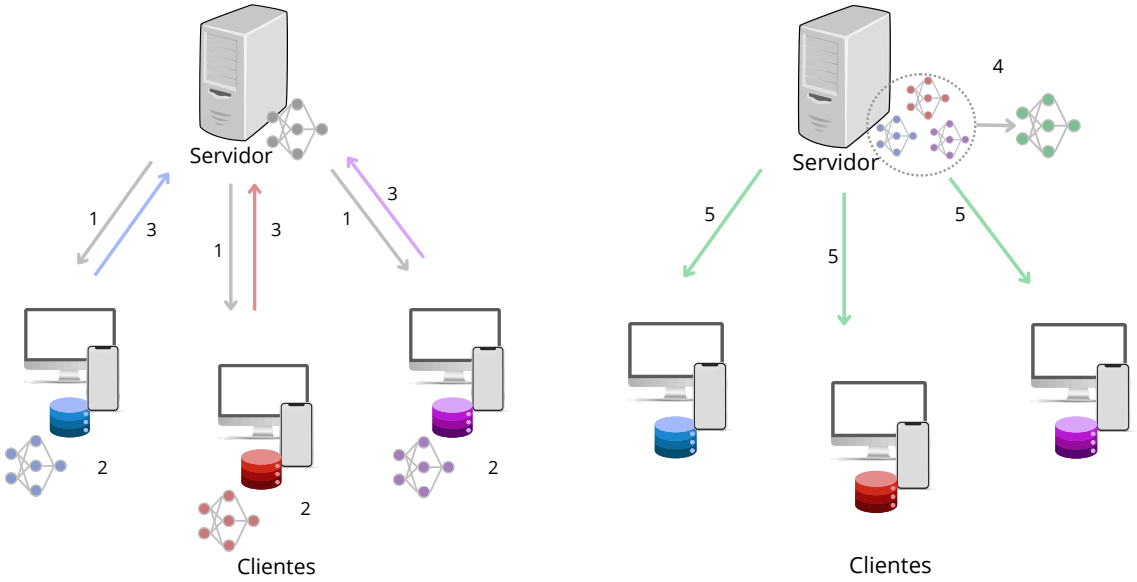
Conforme mencionado no capítulo anterior, um dos tipos de classificação das técnicas de aprendizado de máquina considera a maneira como os clientes treinam seus modelos. O aprendizado federado surge como uma forma de aprendizado colaborativo e é a base para a proposta de seleção de modelos. Os clientes se desligam de federações com servidores de acordo com a percepção do cliente de qual deles oferece o melhor modelo para seu conjunto de dados. A proposta considera o interesse dos clientes em manter um modelo até o fim do treinamento do modelo global.

Este capítulo apresenta um resumo teórico sobre o aprendizado federado e três abordagens propostas para seleção de modelos. A primeira deixa de permanecer em federações menos promissoras a partir dos resultados obtidos em uma rodada de avaliação pré-definida. A segunda utiliza as métricas obtidas até a rodada de avaliação, que possui um valor pré-definido, para estimar o valor futuro da métrica e se desliga de federações consideradas menos promissoras. A terceira realiza o desligamento das federações gradualmente, isto é, não desligando todos os clientes de determinada federação ao mesmo tempo, até que cada cliente mantenha o treinamento em um único modelo. Além disso, abordagens consideram um parâmetro β que define se a diferença entre o maior e o segundo maior valor da métrica de avaliação deve ser o único critério de escolha do melhor modelo. Caso a diferença entre os valores esteja dentro do limite tolerado definido por β, a escolha do modelo baseia-se no servidor de menor latência.

Além disso, este capítulo apresenta os resultados experimentais, incluindo o consumo de memória RAM durante o treinamento, o tempo de treinamento dos modelos locais, a função de custo e os valores da métrica AUC-ROC ao longo das rodadas de comunicação. O impacto das propostas na latência acumulada ao longo das rodadas de comunicação é discutido, assim como os servidores escolhidos por cada cliente. Por fim, os modelos finais obtidos são comparados para as abordagens federada e

centralizada.

5.1 Aprendizado federado tradicional



(a) Treinamento local e envio dos pesos aos servidores.

(b) Agregação dos pesos e atualização dos modelos locais.

Figura 5.1: Aprendizado federado tradicional.

A Figura 5.1 ilustra o funcionamento do treinamento de modelos no aprendizado federado. A Figura 5.1a mostra que o processo inicia-se com o servidor central, que distribui um modelo inicial, representado pela rede neural de cor cinza, para todos os clientes na Etapa 1. Na primeira etapa, o modelo é enviado aos clientes para ser treinado por cada um usando individualmente os seus dados locais. Essas redes treinadas localmente pelos clientes são representadas pelas redes neurais de cores vermelha, roxa e azul na Etapa 2. Em seguida, na Etapa 3, os clientes enviam ao servidor as atualizações dos parâmetros de seus modelos. A Figura 5.1b mostra que essas informações dos modelos locais são utilizadas para que, por um algoritmo de agregação, um novo modelo global seja construído. Nos experimentos apresentados a seguir, a estratégia de agregação *Federated Averaging* (*FedAvg*) é adotada. O FedAvg utiliza uma média ponderada para o cálculo dos parâmetros da rede neural global, em que o peso é proporcional ao número de amostras que cada cliente usa durante o treinamento local. Logo, a média ponderada atribui maior grau de influência na construção do modelo global aos clientes com mais amostras. Essa fase é ilustrada na figura na Etapa 4 e o modelo global é identificado pela rede neural de cor verde. Na Etapa 5, o modelo global é redistribuído aos clientes para mais uma rodada de treinamento com dados locais. As etapas são repetidas até a convergência do modelo

global ou até se alcançar um critério de parada pré-estabelecido.

5.1.1 Seleção de modelo com parâmetro r fixo

A seleção de modelo com o parâmetro r fixo é a primeira proposta apresentada nesta tese. A Figura 5.2 apresenta os passos para que os clientes permaneçam em apenas uma federação considerando uma única rodada de avaliação. A proposta considera um cenário com N servidores, cada um treinando um modelo distinto, e M clientes. O parâmetro r corresponde à rodada de comunicação pré-determinada para avaliar os resultados obtidos pelos clientes e estabelecer qual a federação em que o cliente deve permanecer. Inicialmente, os N modelos são inicializados nos M clientes. Então, os M clientes treinam todos os N modelos com seus dados locais, realizando o teste do modelo local com a métrica de avaliação selecionada. Antes do envio dos pesos para agregação, o algoritmo incrementa em 1 o valor da rodada de comunicação (rc na figura). Enquanto o valor da rodada de avaliação não é atingido, os valores dos pesos das redes são enviados aos servidores para a agregação dos pesos e geração de um novo modelo global. Esse processo ocorre até que se alcance a rodada de avaliação e, após ela, continua até que o número total de rodadas de comunicação seja atingido. A rodada de avaliação é pré-definida antes do início do treinamento, mas o número total de rodadas de comunicação depende de cada cenário experimental, considerando que a Parada Antecipada (*Early Stopping*) é adotada para evitar o sobreajuste dos modelos. O sistema define múltiplos modelos e não assume previamente qual é o melhor.

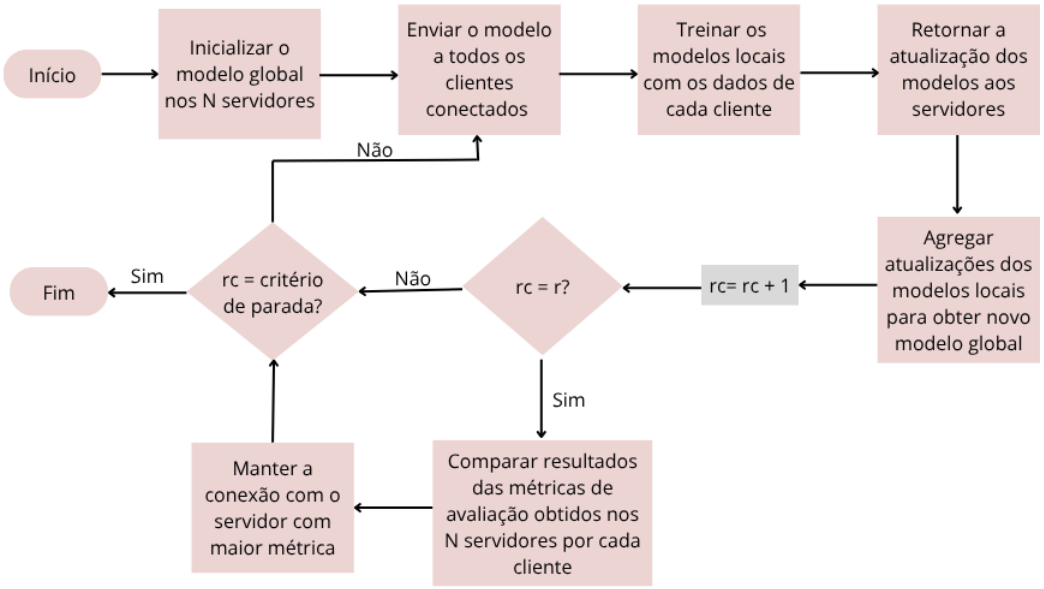


Figura 5.2: Proposta de avaliação das federações com r fixo.

Na rodada de avaliação, os clientes enviam os valores da métrica de avaliação em seu conjunto de teste local ao nó comparador. Esse nó é responsável por realizar o

ranqueamento com os valores da métrica de cada cliente com seu respectivo servidor e indicar em quais federações o cliente deve se manter. Se o nó comparador conclui que o cliente deve se desligar de uma federação, o cliente é responsável por interromper seu treinamento com o servidor e tornar-se indisponível por todo o restante do treinamento. Após a rodada de avaliação, o treinamento continua com os clientes remanescentes em cada servidor.

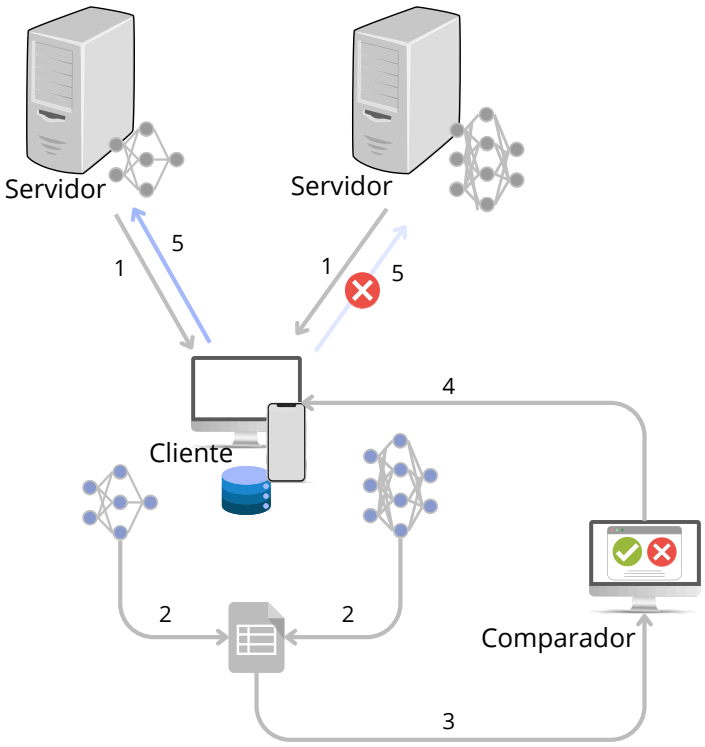


Figura 5.3: Procedimento de desligamento de federações na etapa de avaliação com r fixo.

A Figura 5.3 mostra a proposta considerando a rodada de avaliação do ponto de vista do cliente. Na Etapa 1, os modelos globais são inicializados nos respectivos servidores. Em seguida, os clientes recebem esses modelos e cada cliente treina os N modelos recebidos paralelamente com seus dados locais. O processo de treinamento local e atualização do modelo global ocorre até que se alcance a rodada de avaliação r, em que o resultado obtido no conjunto de testes é enviado ao nó comparador. O nó comparador aguarda até receber todos os resultados dos clientes, uma vez que o tempo de treinamento das redes é distinto, sendo estes valores ranqueados na Etapa 2. Em seguida, o nó comparador indica em qual federação o cliente deve permanecer ou se desligar na Etapa 4 e envia esta informação ao cliente. Por fim, o cliente é responsável por se desligar da federação na Etapa 5. Dessa forma, todos os clientes permanecem até o fim do treinamento de um dos modelos. Sendo este o considerado mais adequado ou promissor na rodada de avaliação.

5.1.2 Seleção de modelos baseada em regressão linear e redução de latência

A segunda proposta de seleção de modelos é baseada em regressão linear e também utiliza a rodada de avaliação r para escolher o modelo mais adequado. Entretanto, em vez de analisar apenas os resultados da rodada r, a proposta baseada em regressão utiliza os resultados da métrica AUC-ROC nas r primeiras rodadas como entrada do algoritmo de regressão linear e estima o valor na próxima rodada. Dessa forma, esse valor previsto é utilizado para definir se determinado deve permanecer em determinada federação ou não. Esta abordagem baseia-se no fato de que valores mais altos em uma rodada específica de avaliação podem mascarar tendências de piora para um determinado cliente.

A Figura 5.4 apresenta as etapas da seleção de modelos baseada em regressão linear. As etapas ilustradas em rosa e cinza são comuns à seleção com o parâmetro r fixo. Por outro lado, a etapa em verde evidencia a mudança entre ambas as propostas conforme aplicado anteriormente.

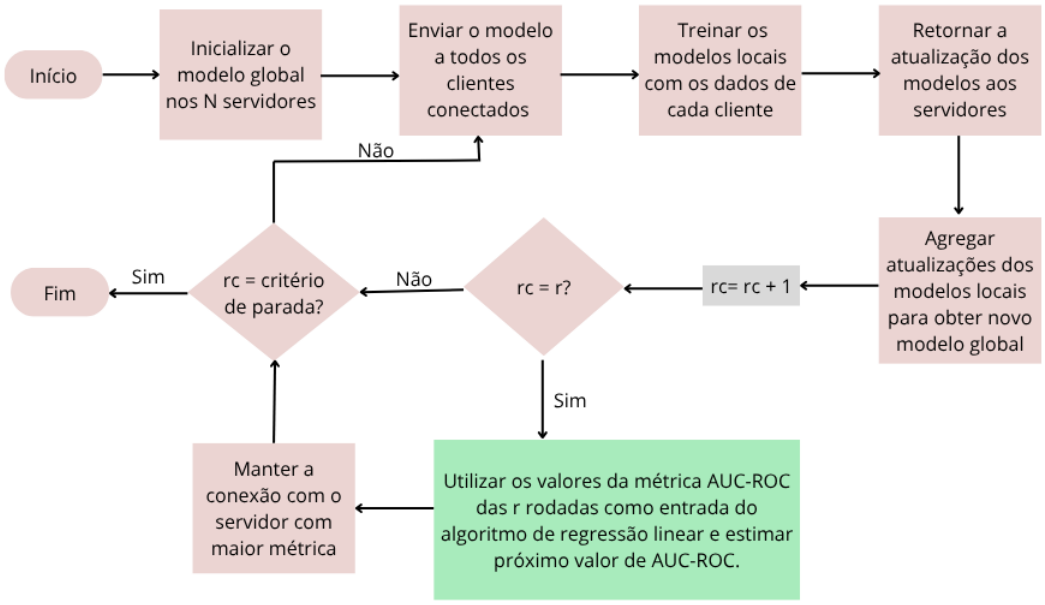


Figura 5.4: Proposta de avaliação das federações baseada em regressão linear.

5.1.3 Seleção de modelo com intervalo de avaliação

A terceira proposta é baseada em um intervalo de avaliação para determinar se o cliente se manterá ou não em uma federação. Nesta proposta o desligamento não ocorre em todos os clientes ao mesmo tempo, mas sim ao longo de rodadas consecutivas, para uma tomada de decisão mais gradual e consistente. A Figura 5.5 apresenta a proposta de seleção de modelo com intervalo de poda ao invés de adotar

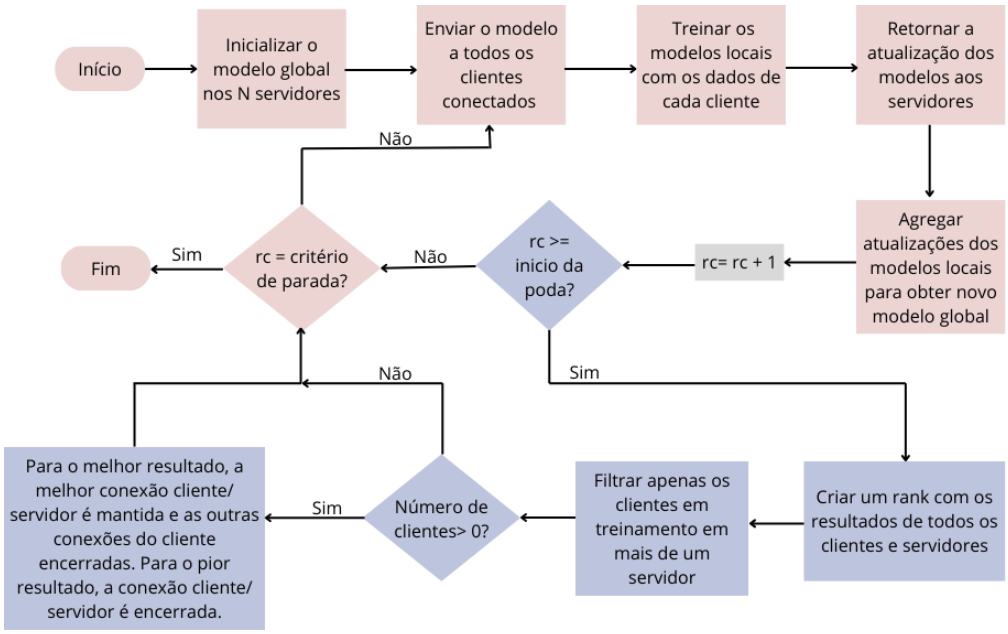
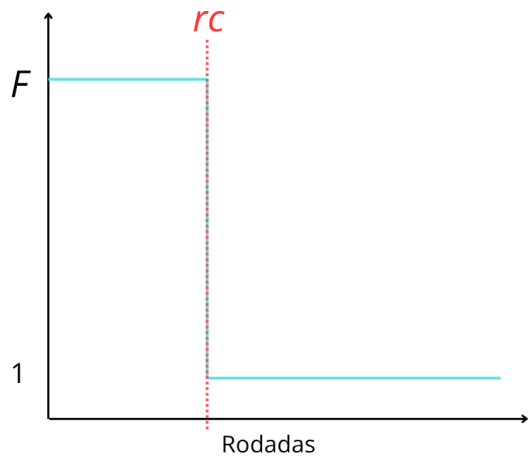


Figura 5.5: Proposta de avaliação das federações com intervalo de poda.

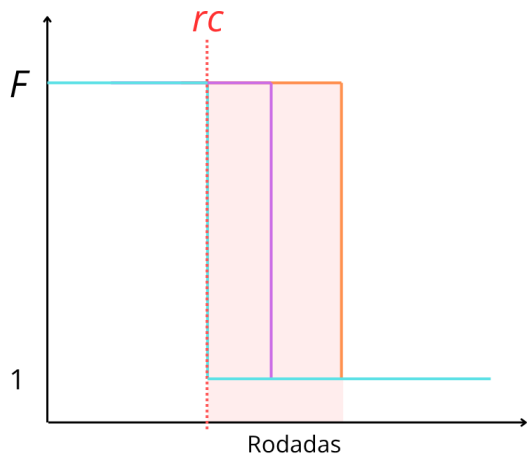
um valor fixo de r. As diferenças entre ambas são evidenciadas pela cor azul do fluxograma. O objetivo, ao adotar um intervalo, é não retirar muitos clientes de uma só vez, o que alteraria bruscamente os pesos a serem agregados na etapa seguinte à poda. Nesta proposta, a rodada de comunicação de início do intervalo é pré-definida e a quantidade de rodadas de avaliação depende do número total de clientes e servidores participantes do treinamento. Uma vez que a rodada de comunicação é igual ou superior ao início do intervalo de poda, os clientes enviam os resultados obtidos nos conjuntos de testes dos modelos locais ao nó comparador. Este nó, assim como na abordagem de r fixo, estabelece uma lista ordenada com todas as métricas obtidas pelos clientes e servidores.

Então, um filtro é aplicado para que apenas os clientes que permanecem em treinamento em mais de um servidor sejam avaliados. Sendo assim, o cliente permanece na federação em que alcançar maior valor de métrica e se desliga das demais. Simultaneamente, o cliente que possui o menor valor de métrica no ranqueamento sairá da respectiva federação em que obteve tal desempenho. Esse procedimento é realizado nas rodadas seguintes enquanto houver clientes treinando mais de um modelo ao mesmo tempo. Uma vez que todos os clientes permanecem com apenas um modelo, o treinamento prossegue normalmente até que o número total de rodadas de comunicação pré-determinado seja atingido.

O número de federações dos clientes e a diferença de funcionamento das propostas são ilustrados na Figura 5.6. Nas propostas baseadas em $rfixo$ e em regressão linear, o número de federações cai de F para 1 assim que se alcança a rodada de comunicação rc definida como rodada de avaliação. Em todos os clientes isto



(a) Federações ao longo das rodadas para todos os clientes das propostas $rfixo$ e da baseada em regressão linear.



(b) Federações ao longo das rodadas para os clientes da proposta com intervalo de avaliação.

Figura 5.6: Número de federações por clientes considerando a rodada de comunicação rc das propostas.

ocorre simultaneamente, ilustrado pela curva azul, conforme a Figura 5.6a. Por outro lado, ao adotar o intervalo de avaliação, cada cliente pode ter sua redução em qualquer momento do intervalo. Desta forma, são ilustrados três clientes distintos na Figura 5.6b representados pelas cores azul, roxa e laranja. O que determina se o cliente será avaliado e desligado das federações, de acordo com o desempenho, é o ranqueamento realizado em cada rodada do intervalo de avaliação.

5.2 Parâmetro β

O parâmetro β é adicionado nas propostas baseadas em regressão linear e em r fixo para mensurar a diferença entre os melhores valores da métrica. Essa avaliação adicional não está presente na proposta baseada em intervalo, pois a essência da construção de avaliações por rodadas consecutivas baseia-se fortemente na avaliação da métrica AUC-ROC apenas.

O pseudocódigo de seleção de modelos para r fixo é apresentado no Algoritmo 1 executado no nó comparador. As informações de todos os clientes C e servidores S participantes dos treinamentos são utilizadas como entrada do algoritmo. Para este experimento são adotados apenas 2 servidores. Além de C e S, os parâmetros β, que permitem analisar a diferença entre as métricas AUC-ROC; a rodada de avaliação r; e o vetor $\mathbf{D_{c,s}}$ com as AUC-ROC das rodadas de treinamento anteriores de cada cliente em um determinado servidor, bem como as latências observadas, são usados como entrada do algoritmo proposto.

Uma vez que um cliente se encontre na rodada de comunicação rc destinada a

Algoritmo 1 Seleção de modelos baseada em $rfixo$ através de métrica de desempenho e latência.

Entradas: C, S, β, r, $\mathbf{D_{c,s}}$
se $rc = r$ **então**
    Aguardar que todos os clientes C enviem os dados de desempenho de treinamento em todos os servidores S
    para cada cliente $c \in C$ **faça**
        para cada servidor $s \in S$ **faça**
            Verificar os valores da métrica $\alpha(c,s)_r$ na rodada de avaliação
        fim para
    fim para
    Ordenar os valores $\alpha(c,S)_r$, sendo $\alpha_1(c,s)_r$ o maior valor da métrica AUC-ROC previsto e $\alpha_2(c,s)_r$ o menor
    se $\alpha_1(c,s)_r \geq \alpha_2(c,s)_r + \beta\alpha_2(c,s)_r$ **então**
        Encerrar treinamento na federação de $\alpha_2(c,s)_r$
    senão
        Encerrar treinamento na federação de maior latência
    fim se
fim se

Algoritmo 2 Seleção de modelos baseada em regressão linear através de métrica de desempenho e latência.

Entradas: C, S, β, r, $\mathbf{D_{c,s}}$
se $rc = r$ **então**
    Aguardar que todos os clientes C enviem os dados de desempenho de treinamento em todos os servidores S
    para cada cliente $c \in C$ **faça**
        para cada servidor $s \in S$ **faça**
            Estimar o próximo valor de métrica $\alpha(c,s)_{r+1}$ em cada servidor através de regressão linear utilizando $\mathbf{D_{c,s}}$
        fim para
    fim para
    Ordenar os valores $\alpha(c,S)_{r+1}$, sendo $\alpha_1(c,s)_{r+1}$ o maior valor da métrica AUC-ROC previsto e $\alpha_2(c,s)_{r+1}$ o menor
    se $\alpha_1(c,s)_{r+1} \geq \alpha_2(c,s)_{r+1} + \beta\alpha_2(c,s)_{r+1}$ **então**
        Encerrar treinamento na federação de $\alpha_2(c,s)_{r+1}$
    senão
        Encerrar treinamento na federação de maior latência
    fim se
fim se

ser a rodada de avaliação r, os valores obtidos da métrica em cada servidor para cada cliente são avaliados. Considerando que $\alpha_1(c)_r$ é a maior AUC-ROC na rodada de avaliação r para o cliente c e $\alpha_2(c)_r$ a segunda maior, ambos os valores são comparados utilizando a Equação 5.1.

$$\frac{\alpha_1(c)_r - \alpha_2(c)_r}{\alpha_2(c)_r} \geq \beta, \tag{5.1}$$

onde apenas o servidor com o maior valor de AUC-ROC é selecionado para a continuação do treinamento do modelo se $\beta = 0,0$. Ao assumir $\beta = 0,0$, as propostas podem ser comparadas com a proposta baseada em intervalo pois significa que a la-

tência não será considerada na avaliação. Porém, quanto maior o valor de β, maior a diferença tolerada entre $\alpha_1(c)_r$ e $\alpha_2(c)_r$ para que a escolha do servidor seja realizada através do critério de latência. Este comportamento é descrito pela Equação 5.2 e generaliza para todas as propostas em que o parâmetro *beta* é utilizado. A F_{esc} significa a federação escolhida, isto é, a federação em que o cliente irá escolher prosseguir com o treinamento. Caso $\frac{\alpha_1(c)_r - \alpha_2(c)_r}{\alpha_2(c)_r} \geq \beta$, a federação escolhida é a com o maior valor da métrica, ou seja, $F_{\alpha_1(c)}$. Caso contrário, será escolhida a federação em que o cliente detectou a latência mínima representada por F_{latmin}.

$$F_{\text{esc}} = \begin{cases} F_{\alpha_1(c)}, & \text{se } \dfrac{\alpha_1(c)_r - \alpha_2(c)_r}{\alpha_2(c)_r} \geq \beta, \\ F_{\text{latmin}}, & \text{caso contrário} \end{cases} \tag{5.2}$$

Note que o objetivo é considerar a latência de comunicação apenas se a diferença de desempenho não for significativa conforme o parâmetro β.

$$\frac{\alpha_1(c)_{r+1} - \alpha_2(c)_{r+1}}{\alpha_2(c)_{r+1}} \geq \beta, \tag{5.3}$$

Na seleção baseada em regressão linear, ao alcançar a rodada de avaliação, o próximo valor de métrica de cada cliente em cada federação é estimado. Considerando que $\alpha_1(c)_{r+1}$ é a maior AUC-ROC prevista para o cliente c e $\alpha_2(c)_{r+1}$ a segunda maior, ambos os valores são comparados utilizando a Equação 5.3. A escolha da federação em que o cliente se manterá até o fim do treinamento segue de acordo com a Equação 5.4.

$$F_{\text{esc}} = \begin{cases} F_{\alpha_1(c)}, & \text{se } \dfrac{\alpha_1(c)_{r+1} - \alpha_2(c)_{r+1}}{\alpha_2(c)_{r+1}} \geq \beta, \\ F_{\text{latmin}}, & \text{caso contrário} \end{cases} \tag{5.4}$$

A diferença entre as equações anteriores consiste na adoção do valor futuro estimado ao invés do valor observado na atual rodada de avaliação.

5.3 Experimentos

Para os experimentos, cada máquina individual nos conjuntos de dados originais corresponde a um número único de cliente do aprendizado federado. A Tabela 5.1 apresenta a correspondência para o conjunto de dados MIMII considerando a SNR igual a 6 dB. Para ilustrar, isso indica que o cliente 1 nos experimentos de aprendizado federado se refere ao ventilador de ID 02 na SNR igual a 6 dB do conjunto de dados original. A Tabela 5.2 apresenta a correspondência para o conjunto de dados ToyADMOS. O número inicia em 48, prevendo que em futuros experimentos as demais relações sinal-ruído do conjunto de dados MIMII podem ser adotadas. Logo,

são utilizados 12 clientes para o conjunto de dados MIMII e todos os 11 clientes do conjunto de dados ToyADMOS.

Tabela 5.1: Correspondência entre as máquinas no conjunto de dados original e suas identificações de clientes adotadas no aprendizado federado para o conjunto de dados MIMII.

Máquina	**ID original**	**ID cliente federado**
Ventilador Industrial	00	0
	02	1
	04	2
Trilho deslizante	00	4
	02	5
	04	6
Válvula	00	8
	02	9
	04	10
Bomba hidráulica	00	12
	02	13
	04	14

O número máximo de rodadas de comunicação adotado é igual a 50. Entretanto, utiliza-se parada antecipada para evitar o sobreajuste. São utilizados três valores para o parâmetro r da rodada de avaliação : 5, 10 e 15. Ou seja, em cada um desses cenários os clientes permanecem em treinamento em mais de um servidor por apenas 10%, 20% e 30% do número máximo total de rodadas, respectivamente. São utilizados 2 servidores nos experimentos, nos quais os modelos são as redes neurais apresentadas no Capítulo 4 e definidas através da etapa de otimização de hiperparâmetros. A avaliação e encerramento da participação de um cliente em uma federação para a proposta com intervalo de poda iniciam-se na rodada 10 e terminam na rodada 16 em ambos os conjuntos de dados. Para o conjunto de dados ToyADMOS, o último cliente se desliga da federação onde alcançou a métrica AUC-ROC mais baixa, uma vez que o número total de clientes é ímpar. A biblioteca Flower é adotada para implementar os cenários de aprendizado federado e as demais etapas foram realizadas utilizando as bibliotecas apresentadas anteriormente na implementação do aprendizado de máquina tradicional.

Os requisitos de aplicações de monitoramento de segurança e alarmes de controle no contexto industrial definidos como entre 5ms e 100ms LU *et al.* (2022) são considerados para simular os valores de latência. Para isso, este intervalo é divido em dois: (A) de 5ms a 52ms, e (B) de 53ms a 100ms. O servidor 1 possui a rede neural com mais camadas e parâmetros treináveis e latências na faixa B e o servidor 2 apresenta latências na faixa A. O objetivo é simular um contexto onde modelos mais complexos podem necessitar de mais recursos computacionais e adotar servidores mais distantes. Dessa forma, o servidor mais complexo tende a ter resultados melhores, mas o servidor mais simples é privilegiado quando a latência é conside-

Tabela 5.2: Correspondência entre as máquinas no conjunto de dados original e suas identificações de clientes adotadas no aprendizado federado para o conjunto de dados ToyADMOS.

Máquina	**ID original**	**ID cliente federado**
Toy Train	1	48
	2	49
	3	50
	4	51
Toy Conveyor	1	52
	2	53
	3	54
Toy Car	1	55
	2	56
	3	57
	4	58

rada. Na inicialização do treinamento, a latência em cada cliente em cada um dos servidores é definida aleatoriamente e os valores observados são apresentados na Tabela 5.3.

A média dos valores simulados e o desvio padrão em cada cenário analisado são apresentados na Tabela 5.3.

Tabela 5.3: Latências usadas na etapa experimental para os diferentes servidores.

Servidor 1			
$\beta = \mathbf{0,0}$	$\beta = \mathbf{0,5}$	$\beta = \mathbf{0,1}$	$\beta = \mathbf{0,15}$
66,865±1,120	67,004±1,109	67,379±1,096	66,88±1,118
Servidor 2			
$\beta = \mathbf{0.0}$	$\beta = \mathbf{0.5}$	$\beta = \mathbf{0,1}$	$\beta = \mathbf{0,15}$
38,166±1,124	37,821±1,102	37,502±1,089	37,945±1,109

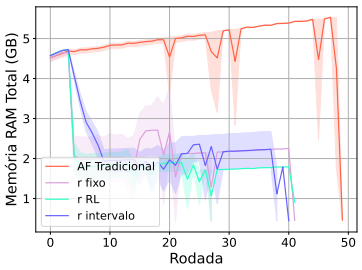
5.4 Resultados

Uma vez que as configurações para os experimentos foram definidas, as abordagens são comparadas tanto considerando os resultados durante o treinamento, quanto nos testes durante as rodadas de comunicação e o modelo final. Os resultados são apresentados com intervalo de confiança de 95% e cada cenário foi realizado por 10 rodadas independentes. Além disso, cada cenário considerou apenas clientes de um mesmo conjunto de dados.

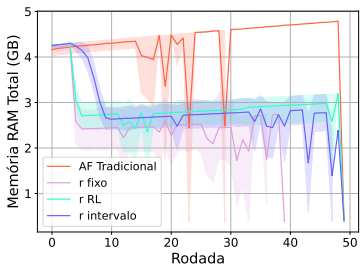
5.4.1 Treinamento dos modelos locais

As Figuras 5.7 e 5.8 apresentam o somatório da memória RAM utilizada durante o treinamento dos modelos em ambos os servidores para os conjuntos de dados MIMII

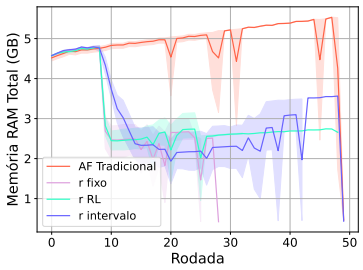
e ToyADMOS, respectivamente. Essa informação é computada para cada cliente ao final do treinamento de seu modelo local na rodada de comunicação correspondente.



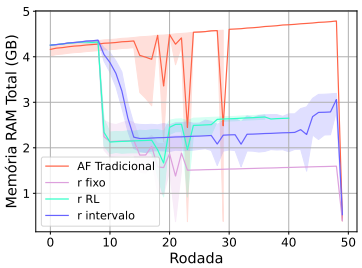
(a) Memória RAM usada durante o treinamento da rede neural 1 para o conjunto de dados MIMII com $r = 5$.



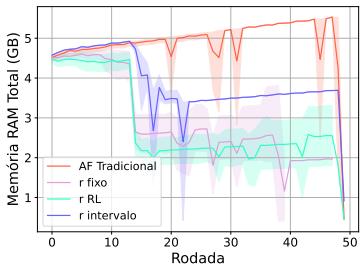
(b) Memória RAM usada durante o treinamento da rede neural 2 para o conjunto de dados MIMII com $r = 5$.



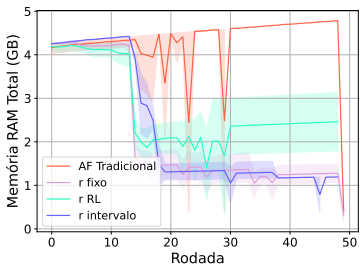
(c) Memória RAM usada durante o treinamento da rede neural 1 para o conjunto de dados MIMII com $r = 10$.



(d) Memória RAM usada durante o treinamento da rede neural 2 para o conjunto de dados MIMII com $r = 10$.



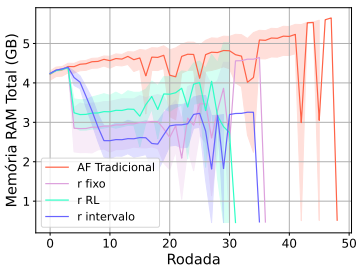
(e) Memória RAM usada durante o treinamento da rede neural 1 para o conjunto de dados MIMII com $r = 15$.



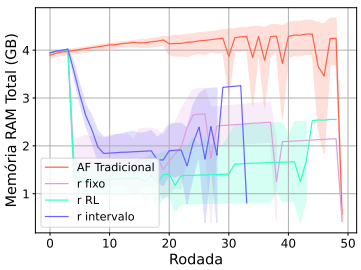
(f) Memória RAM usada durante o treinamento da rede neural 2 para o conjunto de dados MIMII com $r = 15$.

Figura 5.7: Memória RAM utilizado durante o treinamento dos modelos para o conjunto de dados MIMII.

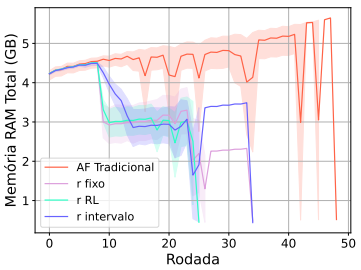
As Figuras consideram o valor de $\beta = 0,0$ como forma de garantir a comparação justa com a proposta baseada em intervalo. A proposta baseada em r fixo é representada pela linha em lilás; a proposta baseada em regressão linear é representada pela linha em verde; e a proposta baseada em intervalo é representada pela linha em azul. É importante ressaltar que apesar do valor máximo de rodadas ser igual a 50 a rodada de parada em cada um dos cenários depende de diversos fatores como, por exemplo, o conjunto ou subconjunto de clientes que permanecem em treinamento. Apesar disso, conforme esperado, as propostas se mantêm abaixo da linha vermelha que representa o aprendizado federal tradicional, ou seja, onde todos os clientes permanecem em treinamento em ambos os servidores do início ao fim. É possível observar a queda da utilização da memória RAM após a rodada de comunicação 5 quando r assume esse valor nas Figuras 5.7a e 5.7b para o conjunto de dados MIMII e nas Figuras 5.8a e 5.8a para o conjunto de dados ToyADMOS. Este comportamento é esperado, uma vez que o número de clientes participando do treinamento é reduzido e pode ser observado para os demais valores de r. Para o conjunto de dados



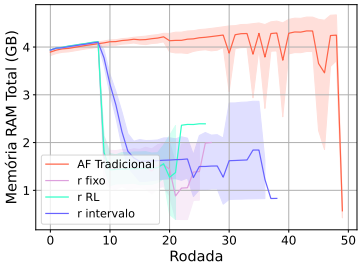
(a) Memória RAM usada durante o treinamento da rede neural 1 para o conjunto de dados TOYADMOS com $r = 5$.



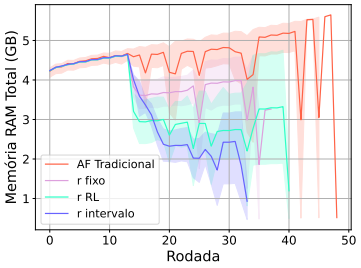
(b) Memória RAM usada durante o treinamento da rede neural 2 para o conjunto de dados TOYADMOS com $r = 5$.



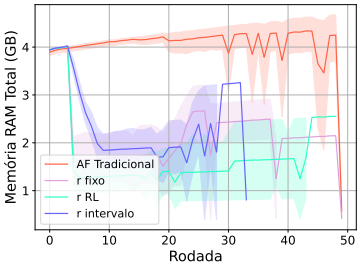
(c) Memória RAM usada durante o treinamento da rede neural 1 para o conjunto de dados TOYADMOS com $r = 10$.



(d) Memória RAM usada durante o treinamento da rede neural 2 para o conjunto de dados TOYADMOS com $r = 10$.



(e) Memória RAM usada durante o treinamento da rede neural 1 para o conjunto de dados TOYADMOS com $r = 15$.



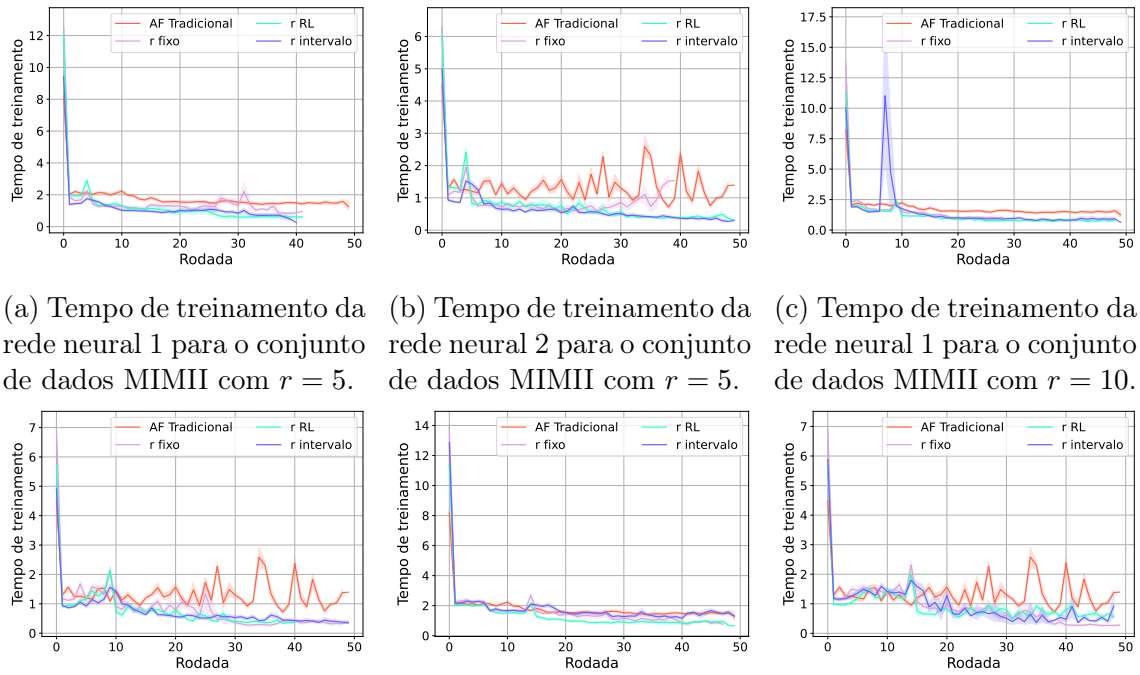
(f) Memória RAM usada durante o treinamento da rede neural 2 para o conjunto de dados TOYADMOS com $r = 15$.

Figura 5.8: Memória RAM utilizada durante o treinamento dos modelos para o conjunto de dados TOYADMOS.

ToyADMOS é possível observar que as propostas tendem a encerrar o treinamento antes do aprendizado federado tradicional em ambos os servidores.

O tempo de treinamento dos modelos é apresentado nas Figuras 5.9 e 5.10. Os resultados mais uma vez são separados pelo valor definido para r e por servidor. As Figuras 5.9a, 5.9c e 5.9e apresentam os resultados para a rede neural/servidor 1 e as Figuras 5.9b, 5.9d e 5.9f para o treinamento da rede neural/servidor 2 para o conjunto de dados MIMII. As propostas necessitam de menos tempo para treinamento dos modelos após a rodada ou intervalo de poda. Entretanto, em r ou no início do intervalo ocorre um pico do tempo necessário de treinamento. O mesmo comportamento é observado durante o treinamento dos modelos com o conjunto de dados ToyADMOS nas Figuras 5.10a, 5.10c e 5.10e apresentam os resultados para a rede neural/servidor 1 e as Figuras 5.10b, 5.10d e 5.10f para o treinamento da rede neural/servidor 2 para o conjunto de dados ToyADMOS.

A função de custo, do ponto de vista dos servidores, ao longo das rodadas de comunicação de treinamento dos modelos para o conjunto de dados MIMII separadas pelos valores distintos do parâmetro β é apresentada nas Figura 5.11 para r



(a) Tempo de treinamento da rede neural 1 para o conjunto de dados MIMII com $r = 5$.

(b) Tempo de treinamento da rede neural 2 para o conjunto de dados MIMII com $r = 5$.

(c) Tempo de treinamento da rede neural 1 para o conjunto de dados MIMII com $r = 10$.

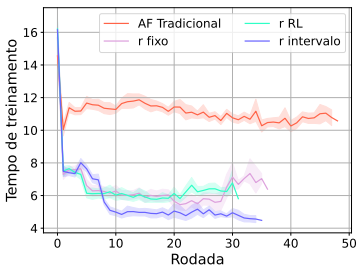
(d) Tempo de treinamento da rede neural 2 para o conjunto de dados MIMII com $r = 10$.

(e) Tempo de treinamento da rede neural 1 para o conjunto de dados MIMII com $r = 15$.

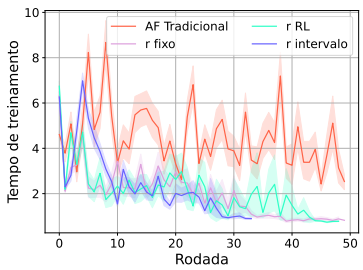
(f) Tempo de treinamento da rede neural 2 para o conjunto de dados MIMII com $r = 15$.

Figura 5.9: Evolução do tempo de treinamento local dos modelos durante as rodadas de comunicação com os servidores para o conjunto de dados MIMII.

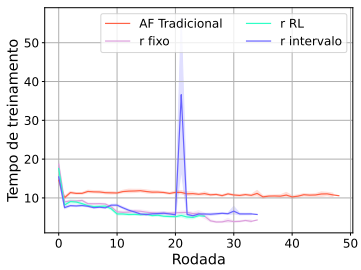
fixo e na Figura 5.12 para a proposta baseada em regressão linear. É importante ressaltar que os pares das propostas ocorrem simultaneamente em ambos os servidores. Isto significa que quando a proposta $r = 5$ é implementada no servidor 1, isto também ocorre no servidor 2 e, assim, ocorre para todas as propostas. Dessa forma, uma curva que permanece com valores mais baixos em um servidor pode ter o comportamento oposto no outro. Isto indica uma possível dificuldade para os modelos obterem uma classificação correta das amostras e pode indicar que os clientes mais promissores permaneceram em um dos servidores. Apesar da expectativa de que os cenários com $\beta = 0,0$ obtenham valores inferiores da função de custo, em alguns cenários analisados ocorre o oposto. Uma das possibilidades é que ao apenas adotar a métrica AUC-ROC como aspecto de escolha, um dos servidores concentre boa parte dos clientes. Desta forma, a convergência dos modelos no outro servidor pode ser comprometida. Os resultados para o conjunto de dados ToyADMOS evidenciam esse comportamento nas Figuras 5.13 e 5.14 e acrescentam outro ponto que deve ser observado: ao adotar valores altos para o parâmetro β, o mesmo fenômeno pode ocorrer, mas, dessa vez, com os clientes concentrados no servidor de menor latência.



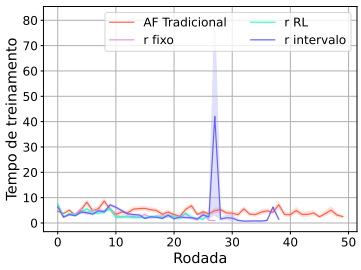
(a) Tempo de treinamento da rede neural 1 para o conjunto de dados TOYADMOS com $r = 5$.



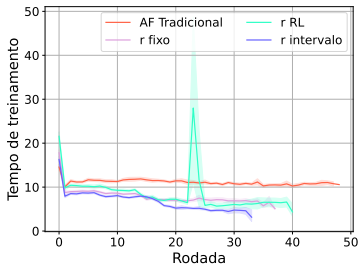
(b) Tempo de treinamento da rede neural 2 para o conjunto de dados TOYADMOS com $r = 5$.



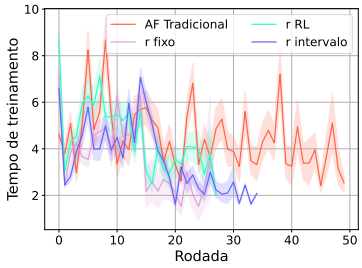
(c) Tempo de treinamento da rede neural 1 para o conjunto de dados TOYADMOS com $r = 10$.



(d) Tempo de treinamento da rede neural 2 para o conjunto de dados TOYADMOS com $r = 10$.



(e) Tempo de treinamento da rede neural 1 para o conjunto de dados TOYADMOS com $r = 15$.



(f) Tempo de treinamento da rede neural 2 para o conjunto de dados TOYADMOS com $r = 15$.

Figura 5.10: Evolução do tempo de treinamento local dos modelos durante as rodadas de comunicação com os servidores para o conjunto de dados TOYADMOS.

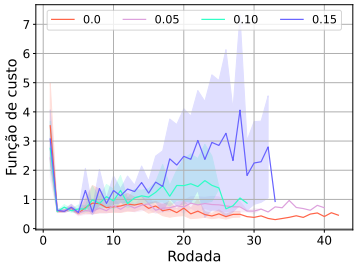
5.4.2 Teste dos modelos

Os resultados da métrica AUC-ROC no teste do modelo são apresentados na Figura 5.15 para o conjunto de dados MIMII e os resultados são separados pelo valor da rodada r definido pelos servidores. Apesar de os resultados das propostas permanecerem próximos entre si ao longo das rodadas de comunicação, o aprendizado federado tradicional permanece inferior na maioria dos cenários.

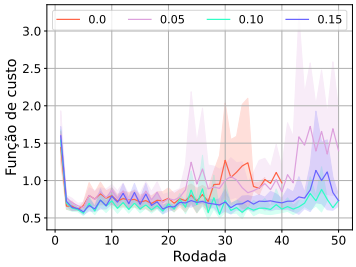
A Figura 5.16 apresenta os resultados para o conjunto de dados ToyADMOS. Nesse contexto, o servidor 1 obtem resultados superiores através do aprendizado federado tradicional. Por outro lado, a rede neural menos complexa obtém valores superiores da AUC-ROC através das propostas e em menos rodadas de comunicação.

5.4.3 Análise de decisão dos clientes

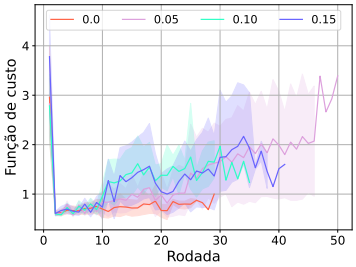
A Figura 5.17 apresenta o impacto da variação do parâmetro β na quantidade de clientes em cada servidor para o conjunto de dados MIMII para as propostas baseadas em r fixo e regressão linear. Quanto maior o valor da rodada de avaliação, mais



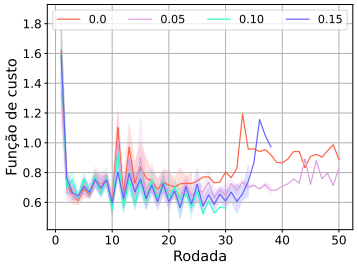
(a) Função de custo da rede neural 1 para o conjunto de dados MIMII para $r = 5$ e $rfixo$.



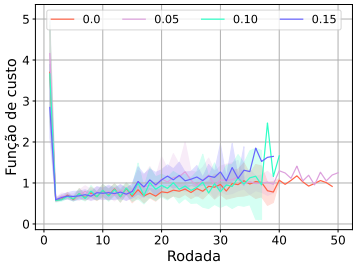
(b) Função de custo da rede neural 2 para o conjunto de dados MIMII para $r = 5$ e $rfixo$.



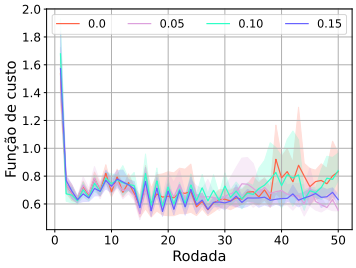
(c) Função de custo da rede neural 1 para o conjunto de dados MIMII para $r = 10$ e $rfixo$.



(d) Função de custo da rede neural 2 para o conjunto de dados MIMII para $r = 10$ e $rfixo$.



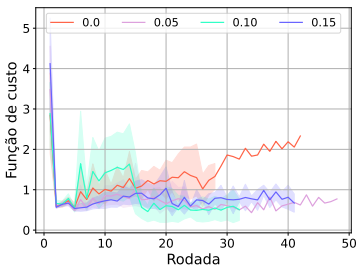
(e) Função de custo da rede neural 1 para o conjunto de dados MIMII para $r = 15$ e $rfixo$.



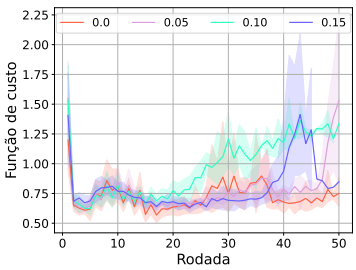
(f) Função de custo da rede neural 2 para o conjunto de dados MIMII para $r = 15$ e $rfixo$.

Figura 5.11: Evolução da função de custo no treinamento dos modelos durante as rodadas de comunicação com os servidores para $rfixo$ e conjunto de dados MIMII.

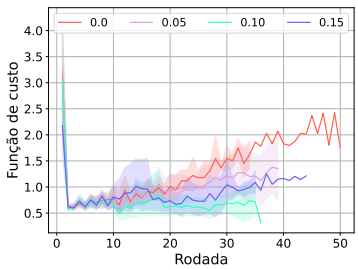
clientes tendem a permanecer no servidor 1. Esse resultado indica que a rede neural mais complexa necessita de mais rodadas de treinamento para que sua métrica AUC-ROC seja superior à obtida pela rede neural com menos parâmetros treináveis. Ao aumentar a tolerância para 5% para avaliar e adotar o servidor com menor latência, isto é, o servidor 2, é possível observar o aumento de clientes no segundo servidor. Ou seja, apesar do servidor 1 apresentar resultados da métrica AUC-ROC superior ao servidor 2, a diferença nesse ponto do treinamento é apenas ligeiramente superior. Para o conjunto de dados ToyADMOS o impacto no aumento do valor de β é ainda maior e apresentado na Figura 5.18. Os resultados da proposta com intervalo de avaliação são apresentados na Figura 5.19. O impacto da quantidade de clientes que permanecem em cada servidor no conjunto de dados MIMII está atrelado à variação do valor r conforme descrito anteriormente. Por outro lado, para o conjunto ToyADMOS, não há impacto direto do treinamento das redes por mais ou menos rodadas na decisão.



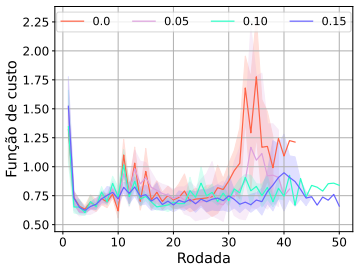
(a) Função de custo da rede neural 1 para o conjunto de dados MIMII para $r = 5$ e rLR.



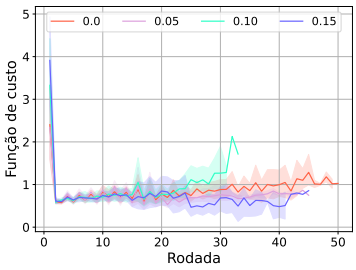
(b) Função de custo da rede neural 2 para o conjunto de dados MIMII para $r = 5$ e rLR.



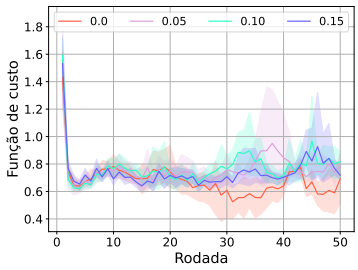
(c) Função de custo da rede neural 1 para o conjunto de dados MIMII para $r = 10$ e rLR.



(d) Função de custo da rede neural 2 para o conjunto de dados MIMII para $r = 10$ e rLR.



(e) Função de custo da rede neural 1 para o conjunto de dados MIMII para $r = 15$ e rLR.

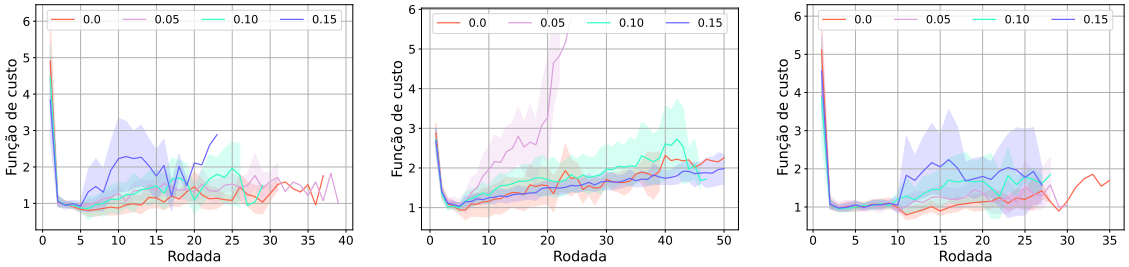


(f) Função de custo da rede neural 2 para o conjunto de dados MIMII para $r = 15$ e rLR.

Figura 5.12: Evolução da função de custo no treinamento dos modelos durante as rodadas de comunicação com os servidores para rLR e conjunto de dados MIMII.

5.4.4 Latência acumulada

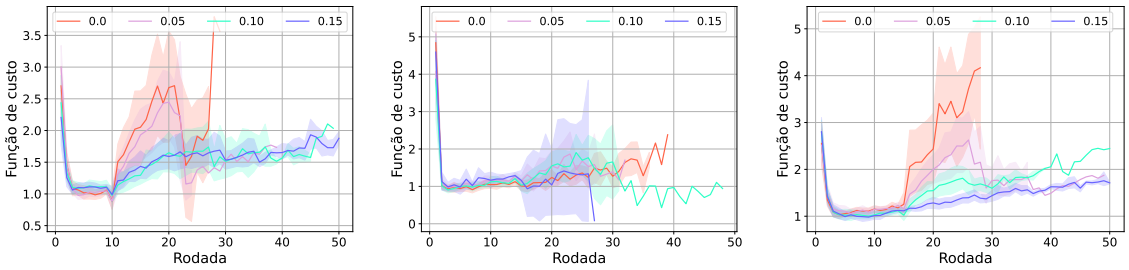
A latência acumulada em segundos durante o treinamento dos modelos é apresentada na Figura 5.20. A latência acumulada observada para todos os valores de β permanece igual até sua respectiva rodada de decisão r. Em todas as variações de r o valor da latência acumulada ao final das 50 rodadas máximas de comunicação é o maior alcançado com $\beta = 0,0$ na maioria dos cenários. Este fato corrobora a análise anterior, indicando que a maioria dos clientes descarta o servidor de menor latência quando a decisão se baseia apenas no valor da métrica, sem considerar o quanto essa métrica difere do valor obtido pelo outro servidor. Conforme o esperado, quanto antes a avaliação é realizada e maior a tolerância entre os valores da métrica AUC-ROC, menor o tempo de latência observado. Dessa forma, além da redução do gasto de recursos em rodadas de comunicação de servidores menos promissores, o modelo final é gerado mais rapidamente.



(a) Função de custo da rede neural 1 para o conjunto de dados TOYADMOS para $r = 5$ e $rfixo$.

(b) Função de custo da rede neural 2 para o conjunto de dados TOYADMOS para $r = 5$ e $rfixo$.

(c) Função de custo da rede neural 1 para o conjunto de dados TOYADMOS para $r = 10$ e $rfixo$.



(d) Função de custo da rede neural 2 para o conjunto de dados TOYADMOS para $r = 10$ e $rfixo$.

(e) Função de custo da rede neural 1 para o conjunto de dados TOYADMOS para $r = 15$ e $rfixo$.

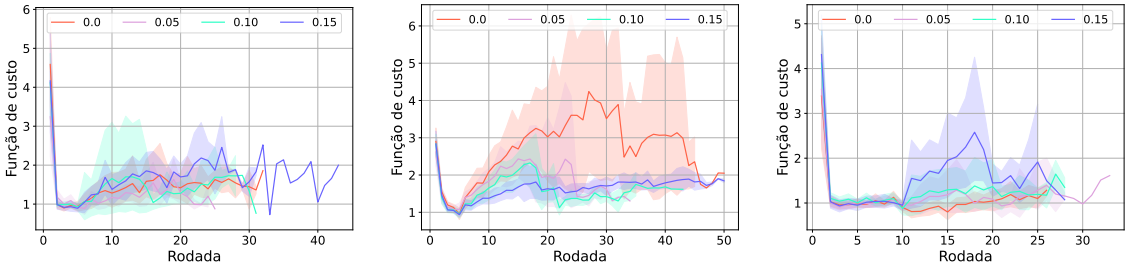
(f) Função de custo da rede neural 2 para o conjunto de dados TOYADMOS para $r = 15$ e $rfixo$.

Figura 5.13: Evolução da função de custo no treinamento dos modelos durante as rodadas de comunicação com os servidores para $rfixo$ e conjunto de dados TOYADMOS.

5.4.5 Avaliação do modelo final

A Figura 5.21 apresenta a métrica AUC-ROC dos modelos finais, separando os resultados obtidos pelos tipos de máquinas. As abordagens baseadas em aprendizado federado são comparadas com as abordagens centralizadas da mesma arquitetura de rede e representadas pelas barras em cinza. Observando os resultados para o conjunto de dados MIMII nas Figuras 5.21a e 5.21b, é possível observar que o aprendizado federado tradicional se mantém com média AUC-ROC abaixo das propostas na maioria dos equipamentos. Este resultado é o esperado, uma vez que os clientes que não auxiliam na convergência do modelo permanecem no treinamento tradicional. As barras correspondentes ao aprendizado centralizado demonstram que os diferentes tipos de equipamentos não auxiliam a convergência do modelo final tanto quanto um modelo específico.

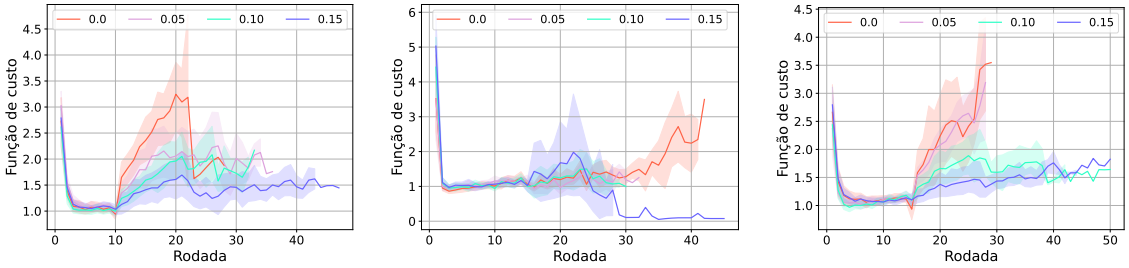
Os resultados obtidos indicam que as propostas analisadas permitem que cada cliente prossiga com o treinamento em uma única federação e alcance em alguns cenários valores da métrica AUC-ROC superiores aos obtidos pelo aprendizado centralizado e federado tradicional. Ao inserir o parâmetro β como fator de decisão,



(a) Função de custo da rede neural 1 para o conjunto de dados TOYADMOS para $r = 5$ e rLR.

(b) Função de custo da rede neural 2 para o conjunto de dados TOYADMOS para $r = 5$ e rLR.

(c) Função de custo da rede neural 1 para o conjunto de dados TOYADMOS para $r = 10$ e rLR.



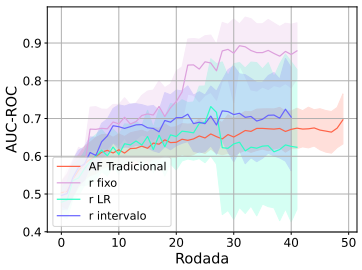
(d) Função de custo da rede neural 2 para o conjunto de dados TOYADMOS para $r = 10$ e rLR.

(e) Função de custo da rede neural 1 para o conjunto de dados TOYADMOS para $r = 15$ e rLR.

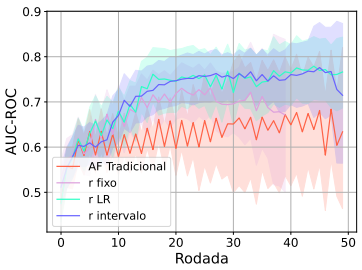
(f) Função de custo da rede neural 2 para o conjunto de dados TOYADMOS para $r = 15$ e rLR.

Figura 5.14: Evolução da função de custo no treinamento dos modelos durante as rodadas de comunicação com os servidores para rLR e conjunto de dados TOYADMOS.

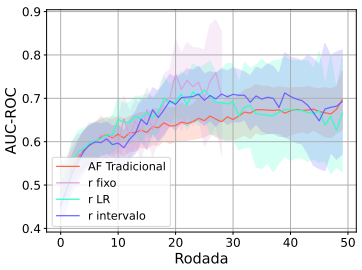
os clientes podem ser mais bem distribuídos entre os servidores, considerando que pequenas diferenças da métrica de desempenho no ponto de avaliação podem não ser o suficiente para determinar se uma federação é realmente a melhor.



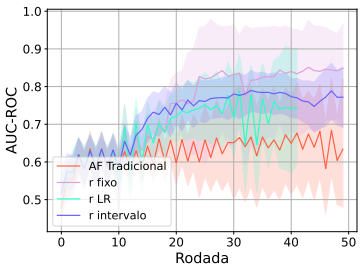
(a) AUC-ROC da rede neural 1 para o conjunto de dados MIMII com $r = 5$.



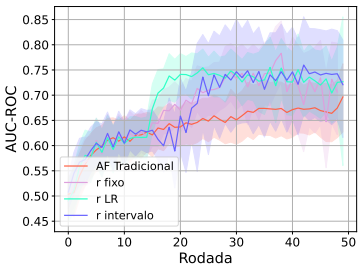
(b) AUC-ROC da rede neural 2 para o conjunto de dados MIMII com $r = 5$.



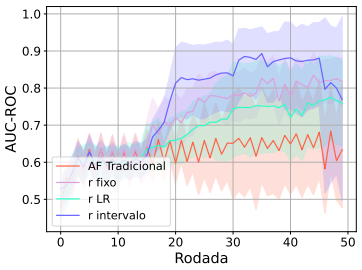
(c) AUC-ROC da rede neural 1 para o conjunto de dados MIMII com $r = 10$.



(d) AUC-ROC da rede neural 2 para o conjunto de dados MIMII com $r = 10$.

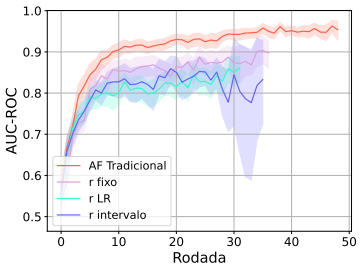


(e) AUC-ROC da rede neural 1 para o conjunto de dados MIMII com $r = 15$.

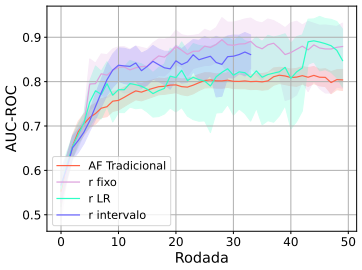


(f) AUC-ROC da rede neural 2 para o conjunto de dados MIMII com $r = 15$.

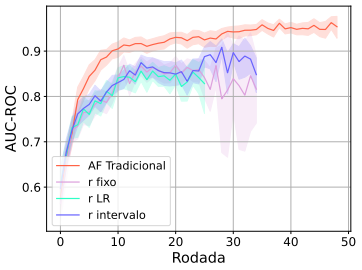
Figura 5.15: Evolução da AUC-ROC dos modelos durante as rodadas de comunicação com os servidores para o conjunto de dados MIMII.



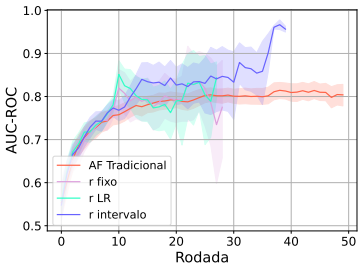
(a) AUC-ROC da rede neural 1 para o conjunto de dados ToyADMOS com $r = 5$.



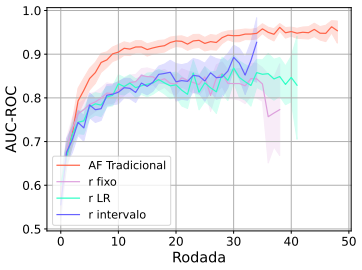
(b) AUC-ROC da rede neural 2 para o conjunto de dados ToyADMOS com $r = 5$.



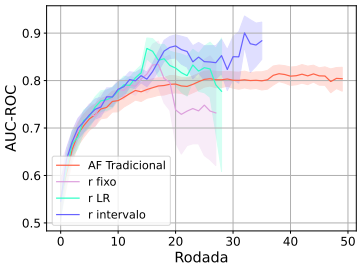
(c) AUC-ROC da rede neural 1 para o conjunto de dados ToyADMOS com $r = 10$.



(d) AUC-ROC da rede neural 2 para o conjunto de dados ToyADMOS com $r = 10$.

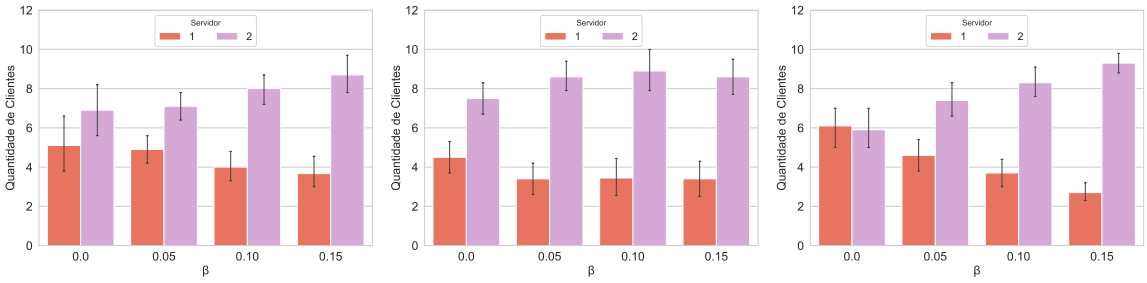


(e) AUC-ROC da rede neural 1 para o conjunto de dados ToyADMOS com $r = 15$.



(f) AUC-ROC da rede neural 2 para o conjunto de dados ToyADMOS com $r = 15$.

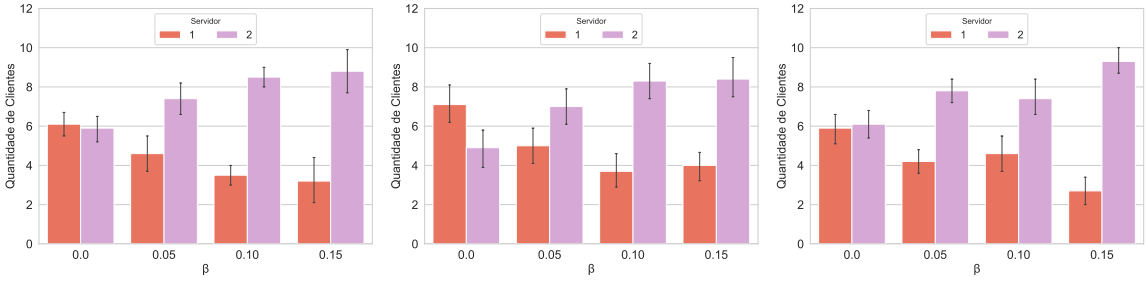
Figura 5.16: Evolução da AUC-ROC dos modelos durante as rodadas de comunicação com os servidores para o conjunto de dados ToyADMOS.



(a) Quantidade de clientes nos servidores conjunto de dados MIMII com $r = 5$ e $rfixo$.

(b) Quantidade de clientes nos servidores conjunto de dados MIMII com $r = 5$ e rLR.

(c) Quantidade de clientes nos servidores conjunto de dados MIMII com $r = 10$ e $rfixo$.

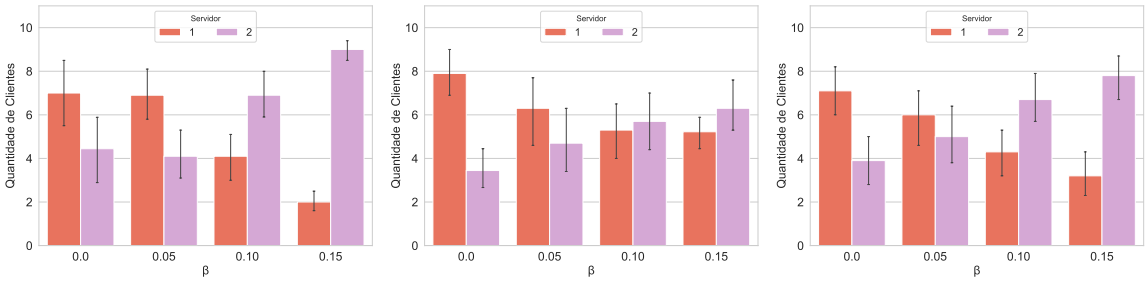


(d) Quantidade de clientes nos servidores conjunto de dados MIMII com $r = 10$ e rLR.

(e) Quantidade de clientes nos servidores conjunto de dados MIMII com $r = 15$ e $rfixo$.

(f) Quantidade de clientes nos servidores conjunto de dados MIMII com $r = 15$ e rLR.

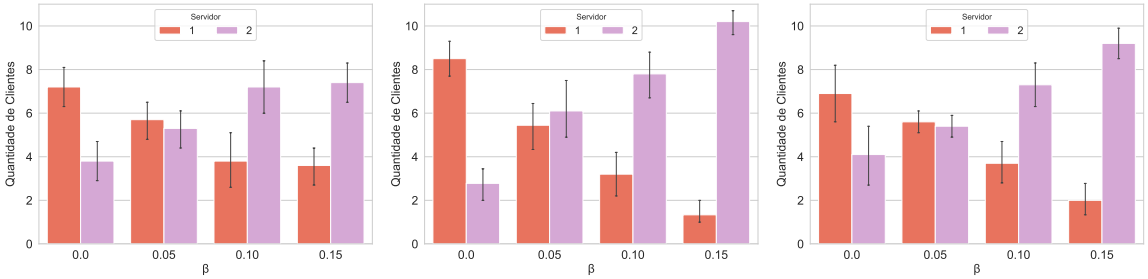
Figura 5.17: Escolha dos clientes pelos servidores nas 10 rodadas independentes para o conjunto de dados MIMII.



(a) Quantidade de clientes nos servidores conjunto de dados TOYADMOS com $r = 5$ e $rfixo$.

(b) Quantidade de clientes nos servidores conjunto de dados TOYADMOS com $r = 5$ e rRL.

(c) Quantidade de clientes nos servidores conjunto de dados TOYADMOS com $r = 10$ e $rfixo$.

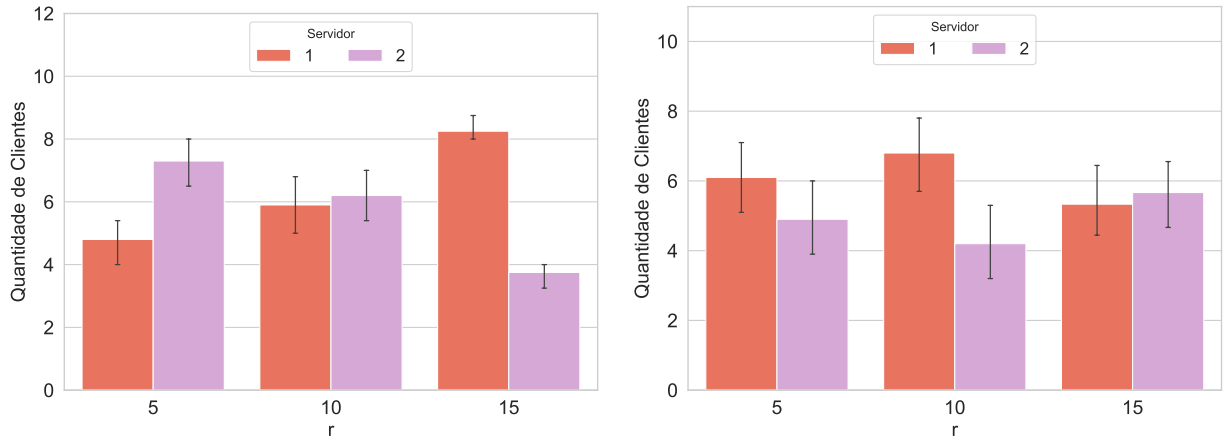


(d) Quantidade de clientes nos servidores conjunto de dados TOYADMOS com $r = 10$ e rLR.

(e) Quantidade de clientes nos servidores conjunto de dados TOYADMOS com $r = 15$ e $rfixo$.

(f) Quantidade de clientes nos servidores conjunto de dados TOYADMOS com $r = 15$ e rLR.

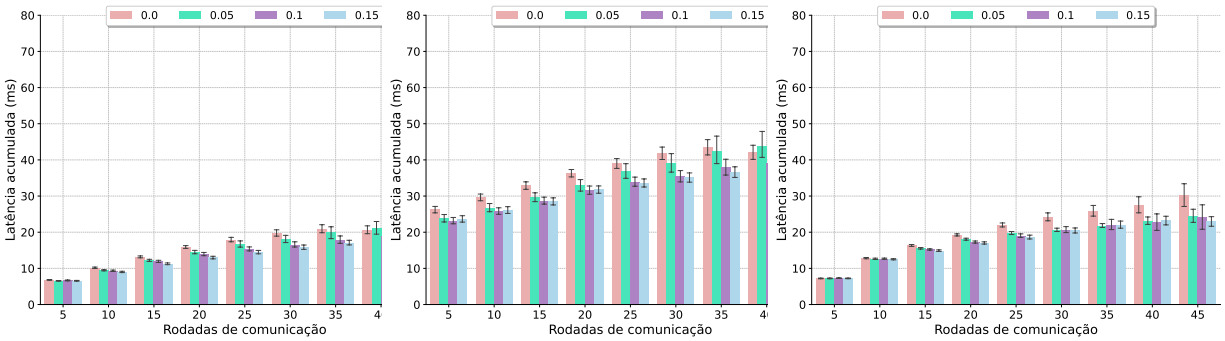
Figura 5.18: Escolha dos clientes pelos servidores nas 10 rodadas independentes para o conjunto de dados TOYADMOS.



(a) Quantidade de clientes nos servidores conjunto de dados MIMII com $rintervalo$.

(b) Quantidade de clientes nos servidores conjunto de dados TOYADMOS com $rintervalo$.

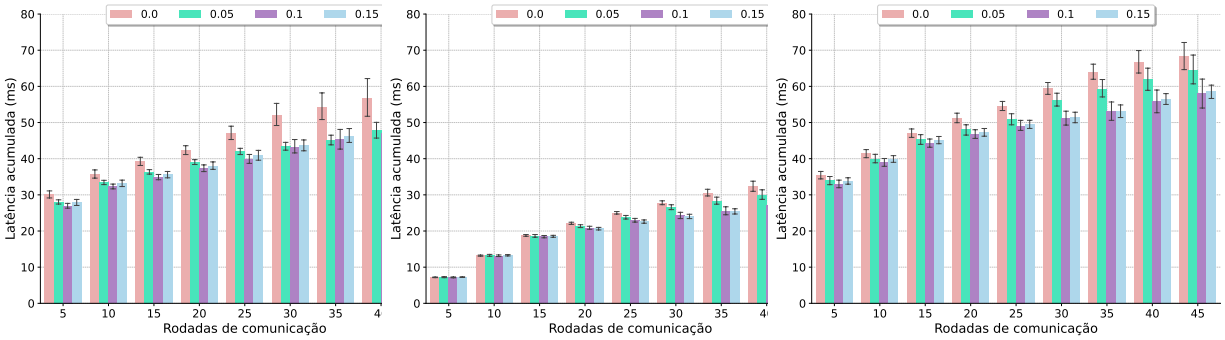
Figura 5.19: Escolha dos clientes pelos servidores nas 10 rodadas independentes para o conjunto de dados TOYADMOS.



(a) Latência acumulada para $r = 5$ para o conjunto de dados MIMII.

(b) Latência acumulada para $r = 5$ para o conjunto de dados ToyADMOS.

(c) Latência acumulada para $r = 10$ para o conjunto de dados MIMII.

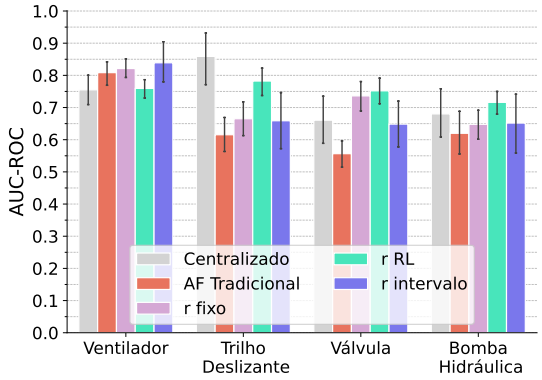


(d) Latência acumulada para $r = 10$ para o conjunto de dados ToyADMOS.

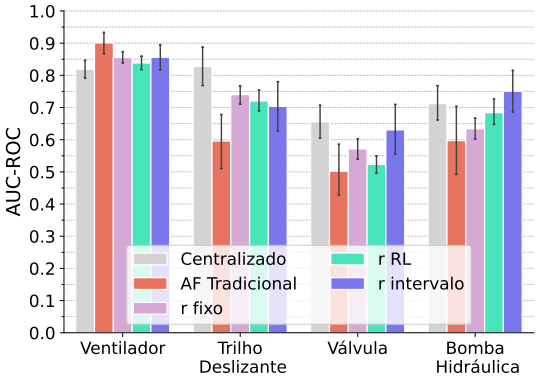
(e) Latência acumulada para $r = 15$ para o conjunto de dados MIMII.

(f) Latência acumulada para $r = 15$ para o conjunto de dados ToyADMOS.

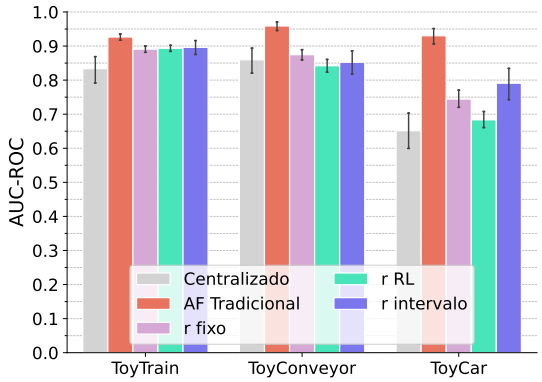
Figura 5.20: Latência acumulada durante as rodadas de comunicação de treinamento para os distintos valores de r e β para ambos os conjuntos de dados.



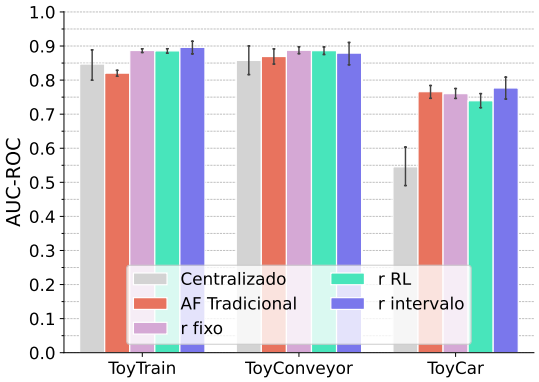
(a) AUC-ROC do modelo final da rede neural 1 para o conjunto de dados MIMII.



(b) AUC-ROC do modelo final da rede neural 2 para o conjunto de dados MIMII.



(c) AUC-ROC do modelo final da rede neural 1 para o conjunto de dados ToyADMOS.



(d) AUC-ROC do modelo final da rede neural 2 para o conjunto de dados ToyADMOS.

Figura 5.21: Comparação por equipamento da AUC-ROC obtida pelos modelos finais de aprendizado federado e centralizado.

Capítulo 6

Conclusão e Trabalhos Futuros

Este trabalho teve como proposta obter os modelos mais adequados de aprendizado de máquina para detecção de defeitos de máquinas industriais a partir de amostras de som, considerando máquinas distintas e a capacidade de um único modelo poder ser adotado por equipamentos distintos. Foram analisados dois tipos de aprendizado de máquina: aprendizado centralizado e aprendizado federado. Em ambos os aprendizados, as características espectrais extraídas das amostras de som foram utilizadas, uma vez que esta abordagem contrasta com as baseadas em espectrogramas por necessitar de menor poder computacional. Isso ocorre, pois os espectrogramas são tratados como imagens e adotam soluções clássicas de classificação de imagens como, por exemplo, redes neurais convolucionais.

Este trabalho utilizou os conjuntos de dados MIMII e ToyADMOS. Do conjunto MIMII foram utilizadas amostras dos quatro equipamentos distintos na SNR = 6 dB, por ser o cenário com menor dificuldade na classificação dos defeitos. Do conjunto ToyADMOS foram utilizadas todas as amostras sonoras de 10 segundos dos três equipamentos. Esta escolha foi baseada na duração das amostras sonoras do conjunto de dados MIMII, uma vez que também são iguais a 10 segundos. Ambos os conjuntos possuem amostras sonoras de equipamentos funcionando segundo o esperado e fora dos padrões. Os dois conjuntos são desbalanceados e possuem quantidade reduzida de amostras anômalas. Uma vez que esta classe é a de maior interesse por indicar possíveis falhas dos equipamentos, foi utilizado um método de sobreamostragem para equilibrar a quantidade de amostras nos dados de treinamento dos modelos.

No aprendizado centralizado foi analisado o método de combinação de algoritmos Super Learner. O método aplicado utilizou cinco algoritmos de classificação binária como base: *AdaBoost*, Naïve Bayes, Máquina de vetores de suporte, K-vizinhos mais próximos e Floresta aleatória. Nesta análise o método Super Learner foi comparado com os algoritmos individualmente e com outras duas redes neurais perceptron de múltiplas camadas. OS hiperparâmetros das redes neurais foram selecionados através do método de otimização bayesiana. A avaliação dos modelos foi realizada

através das métricas AUC-ROC, *recall*, precisão e MCC(*Matthews Correlation Coefficient*). Por um lado, o método Super Learner teve um desempenho melhor que a maioria de seus algoritmos base individualmente nas métricas analisadas. Entretanto, algoritmos como SVM e as redes neurais se destacaram em comparação com o Super Learner. Além disso, no aprendizado centralizado foi analisada a estratégia baseada em classificação multiclasse. Neste tipo de classificação, o modelo é treinado com dados de todos os equipamentos e estados e os rótulos das amostras contêm ambas as informações. O número de classificadores necessários é reduzido em até 83% na classificação multiclasse no conjunto de dados MIMII, além de obter taxas de acerto de detecção de amostras anômalas superiores a 94%. Entretanto, o tempo de treinamento dos modelos multiclasse aumentou consideravelmente em relação aos modelos binários. Porém, um dos pontos ressaltados do aprendizado centralizado é a necessidade de cada máquina/cliente treinar seu próprio modelo. Em um contexto com N clientes são necessários N modelos treinados. Em contrapartida, no aprendizado federado, os N clientes colaboram entre si para gerar um único modelo global.

No aprendizado federado foram analisadas três propostas para seleção de modelo/servidor considerando os resultados intermediários dos clientes. As redes neurais adotadas foram as perceptron de múltiplas camadas (*Multi Layer Perceptron* – MLP) selecionadas no aprendizado centralizado. Na primeira proposta foi utilizado um parâmetro rc para definir uma única rodada de avaliação, em que os resultados da métrica AUC-ROC obtidos por todos os clientes em cada um dos servidores. Este resultado foi salvo por um nó comparador em uma lista ordenada e os clientes permaneceram em treinamento apenas no servidor que alcançou maior métrica na rodada de avaliação r. Na segunda proposta, as rc rodadas iniciais foram utilizadas como entrada do algoritmo de regressão linear para estimar o valor futuro da métrica. O objetivo era que um único ponto analisado não mascarasse uma tendência de redução da métrica avaliada. Na terceira proposta foi utilizado um intervalo de avaliação dos resultados intermediários antes que um cliente se desligasse de determinada federação. A cada rodada do intervalo, os clientes se desligavam de uma federação considerando uma lista ordenada de resultados geral e o pior resultado e o pior modelo do cliente com valor de métrica mais alto. O intervalo prosseguia até que cada cliente estivesse treinando apenas um modelo. Ao analisar as escolhas dos clientes, foi possível identificar que não havia um servidor que fosse o melhor para os clientes em todos os cenários, e sim o servidor mais adequado naquele determinado momento de avaliação, dependendo dos demais clientes e dos pesos do modelo global. Os modelos finais das propostas federadas obtiveram AUC-ROC mais alta que a abordagem federada sem encerramento de conexões, ou seja, com todos os clientes treinando o modelo global. Entretanto, para alguns cenários o modelo de

aprendizado centralizado demonstrou melhores resultados. Novamente, cabe ressaltar que o aprendizado federado fornece um único modelo global em contraposição ao número de modelos centralizados que é igual ao número de clientes.

Além disso, foi inserido o parâmetro β nas duas primeiras propostas, que avaliou a diferença entre o melhor modelo para determinado cliente e o segundo melhor modelo. De acordo com a diferença entre ambos relacionada ao parâmetro β, a latência dos servidores era ou não considerada para definir se o cliente se manteria ou não na federação. Através dos experimentos variando o valor de β, foi possível observar maior balanceamento entre os clientes dos servidores.

Entre os futuros trabalhos pretende-se aumentar o maior número de servidores e inserir diferentes tipos de clientes. Também devem ser adicionados fatores adicionais de desempenho e outras métricas de seleção dos modelos do aprendizado federado além da AUC-ROC, como as métricas de avaliação *recall*, precisão e MCC. Além disso, pretende-se prever situações nas quais um cliente deve adotar um modelo centralizado e não participar da criação do modelo global no aprendizado. Desta forma, clientes que não contribuem para a convergência do modelo poderão desempenhar a tarefa sem prejudicar os demais clientes.

Publicações

- GANTERT, Luana; CAMPISTA, Miguel Elias M. Seleção do Modelo Federado Baseada em Previsão de Desempenho e Tempo de Comunicação. In: XLIII Simpósio Brasileiro de Telecomunicações e Processamento de Sinais (SBrT2025), 2025. DOI: 10.14209/sbrt.2025.1571157155

- GANTERT, Luana; ZEFFIRO, Trevor; SAMMARCO, Matteo; CAMPISTA, Miguel Elias M. Multiclass classification of faulty industrial machinery using sound samples. Engineering Applications of Artificial Intelligence, v. 136, p. 108943, 2024. DOI:10.1016/j.engappai.2024.108943

- GANTERT, Luana; CAMPISTA, Miguel Elias M. Seleção de Modelo de Aprendizado Federado Baseado em Busca e Poda para Detecção de Defeitos Industriais. In: Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC), Maio 2024.: SBC, p. 1106–1119. DOI: 10.5753/sbrc.2024.1548

- SAMMARCO, Matteo; Zeffiro, Trevor; GANTERT, Luana; CAMPISTA, Miguel Elias M. Sound Event Detection Via Pervasive Devices for Mobility Surveillance in Smart Cities. In: 2024 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops), Biarritz, França, 2024. p. 581–586. DOI: 10.1109/PerComWorkshops59983.2024.10503381

- GANTERT, Luana; SAMMARCO, Matteo; DETYNIECKI, Marcin; CAMPISTA, Miguel Elias M. Super Learner Ensemble for Sound Classification using Spectral Features. In: 2022 IEEE Latin-American Conference on Communications (LATINCOM), Rio de Janeiro, Brasil, 2022. p. 1–6. DOI: 10.1109/LATINCOM56090.2022.10000704

Referências Bibliográficas

WANG, S., LIANG, Y., LI, W., et al. "Big Data enabled Intelligent Immune System for energy efficient manufacturing management", *Journal of cleaner production*, v. 195, pp. 507–520, 2018.

GHOBAKHLOO, M. "Industry 4.0, digitization, and opportunities for sustainability", *Journal of cleaner production*, v. 252, pp. 119869, 2020.

ACETO, G., PERSICO, V., PESCAPÉ, A. "A survey on information and communication technologies for industry 4.0: State-of-the-art, taxonomies, perspectives, and challenges", *IEEE Communications Surveys & Tutorials*, v. 21, n. 4, pp. 3467–3501, 2019.

LI, L., OTA, K., DONG, M. "Deep learning for smart industry: Efficient manufacture inspection system with fog computing", *IEEE Transactions on Industrial Informatics*, v. 14, n. 10, pp. 4665–4673, 2018.

BOCHIE, K., GILBERT, M. S., GANTERT, L., et al. "A survey on deep learning for challenged networks: Applications and trends", *Journal of Network and Computer Applications*, v. 194, pp. 103213, 2021.

MCMAHAN, B., MOORE, E., RAMAGE, D., et al. "Communication-efficient learning of deep networks from decentralized data". In: *Artificial intelligence and statistics*, pp. 1273–1282. PMLR, 2017.

LI, L., FAN, Y., TSE, M., et al. "A review of applications in federated learning", *Computers & Industrial Engineering*, v. 149, pp. 106854, 2020.

RAMOS, H. S., MAIA, G., PAPA, G. L., et al. "Aprendizado federado aplicado à internet das coisas", *Sociedade Brasileira de Computação*, 2021.

FU, L., ZHANG, H., GAO, G., et al. "Client selection in federated learning: Principles, challenges, and opportunities", *IEEE Internet of Things Journal*, 2023.

NATESHA, B., GUDDETI, R. M. R. "Fog-based intelligent machine malfunction monitoring system for industry 4.0", *IEEE Transactions on Industrial Informatics*, v. 17, n. 12, pp. 7923–7932, 2021.

GANTERT, L., SAMMARCO, M., DETYNIECKI, M., et al. "A supervised approach for corrective maintenance using spectral features from industrial sounds". In: *2021 IEEE 7th World Forum on Internet of Things (WF-IoT)*, pp. 723–728. IEEE, 2021.

GANTERT, L., SAMMARCO, M., DETYNIECKI, M., et al. "Super learner ensemble for sound classification using spectral features". In: *2022 IEEE Latin-American Conference on Communications (LATINCOM)*, pp. 1–6. IEEE, 2022.

SUEFUSA, K., NISHIDA, T., PUROHIT, H., et al. "Anomalous sound detection based on interpolation deep neural network". In: *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 271–275. IEEE, 2020.

TAO, B., CHEN, C., CHEN, H. "Communication Efficient Federated Learning via Channel-wise Dynamic Pruning". In: *2023 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1–6. IEEE, 2023.

PUROHIT, H., TANABE, R., ICHIGE, K., et al. "MIMII dataset: Sound dataset for malfunctioning industrial machine investigation and inspection". 2019.

KOIZUMI, Y., SAITO, S., UEMATSU, H., et al. "ToyADMOS: A dataset of miniature-machine operating sounds for anomalous sound detection". In: *2019 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pp. 313–317. IEEE, 2019.

GUO, L., LEI, Y., XING, S., et al. "Deep convolutional transfer learning network: A new method for intelligent fault diagnosis of machines with unlabeled data", *IEEE Transactions on Industrial Electronics*, v. 66, n. 9, pp. 7316–7325, 2018.

VAKARUK, S., SIERRA-GARCÍA, J. E., MOZO, A., et al. "Forecasting automated guided vehicle malfunctioning with deep learning in a 5G-based industry 4.0 scenario", *IEEE Communications Magazine*, v. 59, n. 11, pp. 102–108, 2021.

ORRÙ, P. F., ZOCCHEDDU, A., SASSU, L., et al. "Machine learning approach using MLP and SVM algorithms for the fault prediction of a centrifugal

pump in the oil and gas industry", *Sustainability*, v. 12, n. 11, pp. 4776, 2020.

CHAI, Z., ZHAO, C. "Enhanced random forest with concurrent analysis of static and dynamic nodes for industrial fault classification", *IEEE Transactions on Industrial Informatics*, v. 16, n. 1, pp. 54–66, 2019.

BATHELT, A., RICKER, N. L., JELALI, M. "Revision of the Tennessee Eastman process model", *IFAC-PapersOnLine*, v. 48, n. 8, pp. 309–314, 2015.

BUI, K.-H. N., OH, H., YI, H. "Traffic density classification using sound datasets: An empirical study on traffic flow at asymmetric roads", *IEEE Access*, v. 8, pp. 125671–125679, 2020.

MOHAIMENUZZAMAN, M., BERGMEIR, C., MEYER, B. "Pruning vs XNOR-net: A comprehensive study of deep learning for audio classification on edge-devices", *IEEE Access*, v. 10, pp. 6696–6707, 2022.

RASTEGARI, M., ORDONEZ, V., REDMON, J., et al. "Xnor-net: Imagenet classification using binary convolutional neural networks". In: *European conference on computer vision*, pp. 525–542. Springer, 2016.

PAHAR, M., KLOPPER, M., WARREN, R., et al. "COVID-19 cough classification using machine learning and global smartphone recordings", *Computers in Biology and Medicine*, v. 135, pp. 104572, 2021.

JAMIL, S., ROY, A. M. "An efficient and robust phonocardiography (PCG)-based valvular heart diseases (VHD) detection framework using vision transformer (VIT)", *Computers in Biology and Medicine*, v. 158, pp. 106734, 2023.

TAMA, B. A., VANIA, M., KIM, I., et al. "An efficientnet-based weighted ensemble model for industrial machine malfunction detection using acoustic signals", *IEEE Access*, v. 10, pp. 34625–34636, 2022.

HEDEYA, M. A., EID, A. H., ABDEL-KADER, R. F. "A super-learner ensemble of deep networks for vehicle-type classification", *IEEE Access*, v. 8, pp. 98266–98280, 2020.

WEI, N., ZHANG, Q., ZHANG, Y., et al. "Super-learner model realizes the transient prediction of CO2 and NOx of diesel trucks: Model development, evaluation and interpretation", *Environment International*, v. 158, pp. 106977, 2022.

JIANG, Y., WANG, S., VALLS, V., et al. “Model pruning enables efficient federated learning on edge devices”, *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

LIN, R., XIAO, Y., YANG, T.-J., et al. “Federated pruning: improving neural network efficiency with federated learning”, *arXiv preprint arXiv:2209.06359*, 2022.

ALHALABI, B., BASURRA, S., GABER, M. M. “FedNets: Federated Learning on Edge Devices Using Ensembles of Pruned Deep Neural Networks”, *IEEE Access*, v. 11, pp. 30726–30738, 2023.

HUANG, N., DAI, M., WU, Y., et al. “Wireless federated learning with hybrid local and centralized training: A latency minimization design”, *IEEE Journal of Selected Topics in Signal Processing*, v. 17, n. 1, pp. 248–263, 2022.

HU, L., HU, Y., JIANG, L., et al. “Federated learning client selection algorithm based on gradient similarity”, *Cluster Computing*, v. 28, n. 2, pp. 134, 2025.

TRINDADE, S., DA FONSECA, N. L. “Multi-Criteria Scoring for Cluster and Client Selection in Heterogeneous Hierarchical Federated Learning”, *IEEE Internet of Things Journal*, 2026.

CHAKI, J. “Pattern analysis based acoustic signal processing: a survey of the state-of-art”, *International Journal of Speech Technology*, v. 24, n. 4, pp. 913–955, 2021.

GANTERT, L., ZEFFIRO, T., SAMMARCO, M., et al. “Multiclass classification of faulty industrial machinery using sound samples”, *Engineering Applications of Artificial Intelligence*, v. 136, pp. 108943, 2024.

GANTERT, L., CAMPISTA, M. E. M. “Seleçao de Modelo de Aprendizado Federado Baseado em Busca e Poda para Detecçao de Defeitos Industriais”. In: *Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC)*, pp. 1106–1119. SBC, 2024.

ZHENG, F., ZHANG, G., SONG, Z. “Comparison of different implementations of MFCC”, *Journal of Computer Science and Technology*, v. 16, n. 6, pp. 582–589, 2001.

CHAWLA, N. V., BOWYER, K. W., HALL, L. O., et al. “SMOTE: Synthetic minority over-sampling technique”, *Journal of artificial intelligence research*, v. 16, pp. 321–357, 2002.

SUJON, K. M., HASSAN, R. B., TOWSHI, Z. T., et al. "When to use standardization and normalization: Empirical evidence from machine learning models and XAI", *IEEE access*, v. 12, pp. 135300–135314, 2024.

VAN DER LAAN, M. J., POLLEY, E. C., HUBBARD, A. E. "Super learner", *Statistical applications in genetics and molecular biology*, v. 6, n. 1, 2007.

THAKUR, D., BISWAS, S. "Guided regularized random forest feature selection for smartphone based human activity recognition", *Journal of Ambient Intelligence and Humanized Computing*, v. 14, n. 7, pp. 9767–9779, 2023.

CHICCO, D., WARRENS, M. J., JURMAN, G. "The Matthews correlation coefficient (MCC) is more informative than Cohen's Kappa and Brier score in binary classification assessment", *IEEE Access*, v. 9, pp. 78368–78381, 2021.

CHICCO, D., JURMAN, G. "The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation", *BMC genomics*, v. 21, pp. 1–13, 2020.

MCFEE, B., RAFFEL, C., LIANG, D., et al. "Librosa: Audio and music signal analysis in python". In: *Proceedings of the 14th python in science conference*, v. 8, pp. 18–25. Citeseer, 2015.

PEDREGOSA, F., VAROQUAUX, G., GRAMFORT, A., et al. "Scikit-learn: Machine learning in Python", *Journal of Machine Learning Research*, v. 12, pp. 2825–2830, 2011.

LU, Y., YANG, L., YANG, S. X., et al. "An intelligent deterministic scheduling method for ultralow latency communication in edge enabled industrial Internet of Things", *IEEE Transactions on Industrial Informatics*, v. 19, n. 2, pp. 1756–1767, 2022.