NORISVALDO FERRAZ JUNIOR

FedSensor: *framework* de aprendizagem federada voltado para a eficiência energética e segurança de dispositivos IoT ultra-restritos

São Paulo

2022

NORISVALDO FERRAZ JUNIOR

FedSensor: *framework* de aprendizagem federada voltado para a eficiência energética e segurança de dispositivos IoT ultra-restritos

Versão Corrigida

Tese apresentada à Escola Politécnica da Universidade de São Paulo para obtenção do título de Doutor em Ciências.

Área de concentração: Sistemas Eletrônicos

Orientador: Prof. Dr. Sergio Takeo Kofuji

Coorientador: Prof. Dr. Anderson Aparecido Alves da Silva

São Paulo

2022

Este exemplar foi revisado e corrigido em relação à versão original, sob responsabilidade única do autor e com a anuência de seu orientador.

São Paulo, 11 de novembro de 2022

Assinatura do autor: _____

Assinatura do orientador: _____

Catalogação-na-publicação

Ferraz Junior, Norisvaldo
FedSensor: framework de aprendizagem federada voltado para a eficiência energética e segurança de dispositivos IoT ultra-restritos / N. Ferraz Junior -- versão corr. -- São Paulo, 2022.
145 p.

Tese (Doutorado) - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Sistemas Eletrônicos.

1.Internet das coisas 2.Aprendizagem federada 3.Dispositivos IoT ultra restritos 4.Eficiência energética 5.Segurança fim-a-fim I.Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Sistemas Eletrônicos II.t.

Jani

Universidade de São Paulo

ATA DE DEFESA

Ata de defesa de Tese do(a) Senhor(a) Norisvaldo Ferraz Junior no Programa: Engenharia Elétrica, do(a) Escola Politécnica da Universidade de São Paulo.

Aos 27 dias do mês de outubro de 2022, no(a) realizou-se a Defesa da Tese do(a) Senhor(a) Norisvaldo Ferraz Junior, apresenta para a obtenção do título de Doutor intitulada:

"*FedSensor: framework* de aprendizagem federada voltado para a eficiência energética e segurança de dispositivos IoT ultra-restritos"

Após declarada aberta a sessão, o(a) Sr(a) Presidente passa a palavra ao candidato para exposição e a seguir aos examinadores para as devidas arguições que se desenvolvem nos termos regimentais. Em seguida, a Comissão Julgadora proclama o resultad

Nome dos Participantes da Banca	Função	Sigla da CPG	Resultado
Sergio Takeo Kofuji	Presidente	EP - USP	APROVADO
Pedro Frosi Rosa	Titular	UFU - Externo	APROVADO
Pedro Luiz Pizzigatti Corrêa	Titular	EP - USP	APROVADO
Flávio de Oliveira Silva	Titular	UFU - Externo	APROVADO
Augusto José Venâncio Neto	Titular	UFRN - Externo	APROVADO

Resultado Final: APROVADO

Parecer da Comissão Julgadora *

Eu, Elias Alves de Almeida _____, lavrei a presente ata, que assino juntamente com os(Senhores(as). São Paulo, aos 27 dias do mês de outubro de 2022.

Pedro Frosi Rosa

Pedro Luiz Pizzigatti Corrêa

Flávio de Oliveira Silva

Augusto José Venâncio Neto

Sergio Takeo Kofuji
Presidente da Comissão Julgadora

* Obs: Se o candidato for reprovado por algum dos membros, o preenchimento do parecer é obrigatório.

A defesa foi homologada pela Comissão de Pós-Graduação em _____ e, portanto, o(a) aluno(a) _____ j ao título de Doutor em Ciências obtido no Programa Engenharia Elétrica - Área de concentração: Sistemas Eletrônicos.

_____
Presidente da Comissão de Pós-Graduação

Prof. Dr. Oswaldo Horikawa
Presidente da Comissão de Pós Graduação

AGRADECIMENTOS

Agradeço ao Senhor Deus, em nome de Jesus, pela oportunidade de ter vida, saúde, oportunidade e ter me dado a graça para chegar até este momento.

Aos meus pais pelo incentivo para iniciar essa longa jornada, e pelos ensinamentos desde cedo!

Um agradecimento especial à minha esposa, que acompanhou diariamente as alegrias, dificuldades, vitórias em cada passo, e pela paciência de chegar até esse momento. Agradeço às minhas filhas, pois acompanharam, incentivaram e participaram de todo esse processo.

À todos os meus familiares que, mesmo os que moram em outra cidade e estando fisicamente distantes, não deixaram de incentivar!

Ao meu orientador, o Prof. Dr. Sergio Takeo Kofuji pela oportunidade de ingresso nesse imenso desafio de pesquisa e pelas orientações e conduções durante o processo.

Ao meu co-orientador, amigo em todas as horas, e irmão Anderson, um profundo agradecimento pelo incentivo diário durante os períodos difíceis. Sempre presente, dando apoio e um grande incentivador e motivador: muito obrigado!

Aos membros da Banca, os Professores Augusto, Pedro Frosi, Pedro Pizzigatti e Flavio, um agradecimento especial pelas importantes contribuições para enriquecimento do trabalho.

Aos membros do PAD que acompanharam a minha trajetória, cada um com a sua parcela de apoio.

À todos os demais que acompanharam essa jornada e que me apoiaram e torceram por mim.

RESUMO

FERRAZ JUNIOR, N. **FedSensor: *framework* de aprendizagem federada voltado para a eficiência energética e segurança de dispositivos IoT ultra-restritos**. 2022. 145 f. Tese (Doutorado) - Escola Politécnica da Universidade de São Paulo, São Paulo, 2022.

Aplicações da Internet of Things (IoT) são utilizadas em vários domínios, como a Indústria 5.0, cidades inteligentes, entre outros, e são compostas por dispositivos, os quais fornecem grande volume de dados. Diferentes dispositivos podem ser utilizados, mas aqueles que se movimentam com frequência e precisam operar por vários dias ou meses sem a substituição das baterias ampliam a área de atuação das aplicações IoT - esses são os dispositivos IoT ultra-restritos (com severas restrições em processamento, memória, energia e tamanho da carga útil). Os dados advindos desses dispositivos viabilizam a tomada de decisão inteligente, esta que resulta da aplicação de modelos de aprendizagem de máquina - *machine learning* (ML). Nesse cenário, ao se considerar a privacidade, o isolamento dos dados no mesmo ambiente inteligente é fundamental. Por isso, a aprendizagem federada - *federated learning* (FL) - permite a realização do treinamento distribuído de um modelo de ML sem que os dados sejam transmitidos da *Edge* para o núcleo da nuvem. Contudo, nos ambientes tradicionais de FL, o gerenciador conhece todos os dispositivos, o que se mostra inseguro. Ainda, os dispositivos ultra-restritos utilizados em sistemas de missão crítica requerem que suas baterias mantenham seu padrão de vida útil com a aplicação de inteligência na tomada de decisão. Outro fator comum nesses dispositivos com severas restrições é o envio de medições anômalas. Diante do exposto, este trabalho apresenta o FedSensor, um *framework* de FL em redes IoT baseadas em sensores e atuadores, que considera a cooperação entre nuvem e *Edge* para a geração de modelos de ML globais viáveis para utilização em dispositivos IoT ultra-restritos. Neste trabalho avalia-se o FedSensor com relação ao consumo de energia e detecção de medições anômalas. O FedSensor propicia a anonimidade dos dispositivos (que não são conhecidos pelo gerenciador e apenas controlado pelos participantes). Além disso, o FedSensor mantém a característica de vida útil de bateria dos dispositivos, mesmo adicionando a inteligência artificial para a tomada de decisão. Os resultados do FedSensor apontam que o maior fator que reduz a vida útil dos dispositivos é o número de desfechos do modelo de ML global em conjunto com a frequência de realização de inferências, e não o volume de mensagens recebidas contendo o modelo de ML global. Em cenários de severa utilização, a média da redução da vida útil das baterias é de 38,59% (em relação ao dispositivo em descanso) e em casos de utilização não severa, a redução média da vida útil é de 2,88%. Por isso, utilizar dispositivos IoT ultra-restritos em arquiteturas de FL, que é um desafio apresentado no estado-da-arte, é viável com o uso do FedSensor, principalmente quando a realização de inferências não precisa ser realizada com intervalo de tempo igual ou menor que dois segundos. Por fim, identificam-se participantes que contém medições anômalas advindas dos dispositivos IoT, ao se observar os resultados da função de custo federado dos modelos de ML, ao se concluir o treinamento federado.

Palavras-chave: Internet das Coisas. Aprendizagem federada. Dispositivos IoT ultra-restritos. Eficiência energética. Segurança e privacidade em IoT.

ABSTRACT

FERRAZ JUNIOR, N. **FedSensor: federated learning framework focused on security and energy-efficiency of ultra-low-power IoT devices**. 2022. 145 f. Tese (Doutorado) - Escola Politécnica, Universidade de São Paulo, São Paulo, 2022.

IoT applications exists in various domains, such as Industry 5.0, smart cities, among others, and are composed of devices, which provide a large amount of data. Among the plethora of devices, there is the ones that are mobile and need to operate for several days or months without replacing the batteries. These are the ultra-low-power IoT devices (with severe restrictions on processing, memory, power, and payload). The data from these devices enable intelligent decision-making, which results from Machine Learning (ML) models. However, considering privacy, data must be isolated in its domain. In this sense, Federated Learning (FL) allows a distributed training of an ML model without data being transmitted from Edge to the core of the cloud. However, in traditional FL environments the manager knows all devices, which is insecure. Also, the ultra-low-power devices used in mission-critical systems require that their batteries maintain their standard of life with the application of intelligence in decision-making. Anomalous measurements collected by the sensors and sent by these devices are common in wireless sensor and actuator networks. Given the above, this work presents FedSensor, an FL framework in sensor-based IoT networks, which considers the cooperation between cloud and Edge to generate viable global ML models for use in ultra-low-power IoT devices. In this work, we evaluate FedSensor in terms of energy consumption and detection of anomalous measurements. FedSensor provides devices' anonymity (which is not known by the manager and is only controlled by the participants). In addition, FedSensor maintains the device's battery life characteristic, even adding artificial intelligence to decision-making. The FedSensor results show that the most important factor that reduces the lifetime of devices is the number of outcomes of the global ML model together with the frequency of inferences, and not the volume of messages received containing the global ML model. In severe usage scenarios, the average battery life reduction is 38.59% (compared to the device in idle), and in non-severe use cases, the average lifespan reduction is 2.88%. Therefore, using ultra-low-power IoT devices in FL architectures (a challenge presented in the state-of-the-art) is feasible with the use of FedSensor; especially when inferences do not need to be performed with an equal time interval or less than two seconds. Finally, we identify participants with anomalous measurements, by observing the results of the federated cost function of the participants' ML models, when completing the federated training.

Keywords: Internet of Things. Federated learning. Ultra-low-power IoT devices. Energy efficiency. Security and privacy in IoT.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

6LoWPAN	IPv6 over Low-Power Wireless Personal Area Networks
AIoT	Interligência artificial das coisas (Artifficial Intelligence of Things)
AMQP	Advanced Message Queuing Protocol
CO	Monóxido de carbono
CO_2	Dióxido de carbono
CoAP	Constrained Application Protocol
DDS	Data Distribution Service
ECOD	Empirical Cumulative distribution functions for Outlier Detection
iForest	Isolation Forest
FL	Aprendizagem federada (Federated Learning)
IEEE	Institute of Electrical and Electronics Engineers
IoT	Internet das Coisas (Internet of Things)
IQAr	Índice de Qualidade do Ar
IVP	Índice de Variabilidade do Participante
J	Joule
JSON	JavaScript Object Notation
LoRa	Long Range
LPM	Low Power Mode
LPWAN	Low-Power Wide Area Network
LWM2M	LightWeight Machine-to-Machine
ML	Aprendizagem de máquina (Machine Learning)

MQTT	Message Queue Telemetry Transport
MTU	Maximum Transfer Unit
mJ	Milijoule
O_3	Ozônio
OMA	Open Mobile Alliance
PM	Material particulado (Particulate Matter)
TX	Transmissão
RSSF	Redes de Sensores Sem Fio
RX	Recepção
SBC	Single Board Computers
USP	Universidade de São Paulo
XMPP	eXtensible Messaging and Presence Protocol

SUMÁRIO

1 INTRODUÇÃO

A Internet of Things (IoT) é uma arquitetura na qual os domínios ou mercados verticais, como a Indústria 5.0, as cidades e os demais ambientes inteligentes fornecem um grande volume de dados proveniente dos sensores dos dispositivos/equipamentos (GUPTA et al., 2020; AL-FUQAHA et al., 2015). É comum que dispositivos IoT atuem em diferentes aplicações e possuam vários sensores embarcados. Esses dispositivos podem ser: de alta capacidade (como smartphones ou Single Board Computers (SBC) como o Raspberry Pi), restritos (que normalmente se comunicam usando IEEE 802.11, como o ESP-32) ou ultra-restritos. Os dispositivos IoT ultra-restritos têm baixíssima capacidade de processamento (são baseados em microcontroladores e não CPU), memória (até 512KB), MTU (até 127 Bytes), e energia (na casa dos miliwatts). Os dispositivos IoT ultra-restritos são aqueles utilizados nas Redes de Sensores Sem Fio (RSSF), como a IPv6 over Low-Power Wireless Personal Area Networks (6LoWPAN) e a Low-Power Wide Area Network (LPWAN), onde realizam monitoramento e atuação em ambientes inteligentes a partir das medições de seus sensores embarcados ou de comandos recebidos.

Independentemente dos dispositivos IoT utilizados, os dados por eles fornecidos podem ser utilizados por modelos de Machine Learning (ML) para subsidiar a tomada de decisão por governantes, cidadãos e a comunidade científica. Porém, aplicações tradicionais de ML exigem um servidor central (normalmente na nuvem) que tenha posse dos dados, inviabilizando o treinamento (caso os dados não possam ser entregues ao servidor central para treinamento, ou exigindo que os dados deixem seu ambiente inteligente para treinamento com a garantia do servidor central de que os dados serão utilizados exclusivamente para esse fim, exigindo mecanismos de segurança adicionais para garantia da privacidade (LU et al., 2020). A transmissão desse volume de dados dos dispositivos para a nuvem pode causar atrasos, tanto no tráfego quanto no processamento. Adicionalmente, segundo Lu et al. (2020), os dados de uma organização específica (como um único hospital, por exemplo) podem ser similares entre si e não diversificados o suficiente, o que pode resultar em modelos sobreajustados. Mesmo a utilização de recursos como a seleção das melhores variáveis resposta, o modelo de ML se torna enviesado em virtude da similaridade dos dados de uma organização específica. Nesse sentido, a descentralização do processamento

dos dados para uma camada mais próxima dos dispositivos, como a *Edge* é necessária e aplicada nos trabalhos de Ye et al. (2020), Saha, Misra e Deb (2021), Merenda, Porcaro e Iero (2020) e Foukalas e Tziouvaras (2021).

Uma *Federated Learning* (FL), segundo McMahan et al. (2017), se refere à união da capacidade de treinamento de modelos de ML fornecida pela computação em nuvem com a crescente capacidade de processamento dos dispositivos móveis. Isso permite atingir dois objetivos: a privacidade dos dados e o aumento da acurácia nos modelos de ML. A FL considera a existência de um modelo de ML global, gerado em um servidor central na nuvem (denominado de gerenciador). O gerenciador envia o modelo de ML global ao cliente (normalmente um dispositivo denominado de participante), que por sua vez coleta os dados e gera um modelo de ML local. O participante, então, envia esse modelo local para o gerenciador. Essa iteração ocorre até que se atinja a acurácia desejada ou que haja convergência do modelo. Neste processo, os dados brutos não são enviados do participante para o gerenciador, conforme apresentam Niknam, Dhillon e Reed (2020), Lim et al. (2020), Wang et al. (2019), Chen et al. (2021) e Feraudo et al. (2020).

Por isso, observa-se que a FL fornece privacidade dos dados coletados pelos participantes, pois os clientes transmitem apenas o modelo de ML local. O objetivo do gerenciador é realizar uma agregação dos modelos de ML locais com o intuito de aumentar a acurácia do modelo de ML global (MCMAHAN et al., 2017). Contudo, mesmo não havendo perda de privacidade dos dados, há perda de privacidade dos dispositivos que participam da arquitetura, pois o servidor na camada nuvem tem acesso direto aos clientes.

A FL, portanto, viabiliza o aprendizado colaborativo de um modelo de ML para os equipamentos participantes. Para viabilizar a inclusão de dispositvos IoT restritos (mas não os ultra-restritos), surge a Edge Intelligence (ZHOU et al., 2019), que permite aos dispositivos dividir com a camada Edge a tarefa de treinamento colaborativo de modelos de ML em ambientes FL, mantendo as características de privacidade dos dados (YE et al., 2020; MERENDA; PORCARO; IERO, 2020; SAHA; MISRA; DEB, 2021; ZHOU et al., 2020; FOUKALAS; TZIOUVARAS, 2021; ZHOU et al., 2019).

Contudo, uma característica encontrada na Edge Intelligence é que ainda ocorre (mesmo que parcialmente) o treinamento dos modelos de ML locais dentro dos dispositivos - o que é um impeditivo para o uso de dispositivos IoT ultra-restritos em arquiteturas de FL,

segundo (ZHOU et al., 2019). Outro fator a ser considerado é a Maximum Transmission Unit (MTU) das redes IoT baseadas em sensores e atuadores (que varia entre 127 e 256 Bytes), a qual é insuficiente para o tráfego de complexos modelos de ML, que podem exigir MegaBytes (MB) de tamanho (XIA et al., 2021).

Por isso, neste trabalho apresenta-se uma proposta para aplicar a FL às redes IoT baseadas em sensores e atuadores, e com isso fornecer inteligência aos dispositivos IoT ultra restritos, permitindo a tomada de decisão no próprio dispositivo. Nessa proposta, a camada em nuvem não tem acesso aos dispositivos (fornecendo privacidade e anonimidade). Adicionalmente, os dispositivos não precisam aguardar a inferência realizada pelo servidor central na nuvem para receber um comando para tomar uma ação. Além disso, propõe-se utilizar dispositivos IoT ultra-restritos pois eles fornecem alta eficiência energética, não somente para os dispositivos, mas também para a arquitetura de FL como um todo.

1.1 Motivação

As primeiras propostas para a tomada de decisão autônoma dos dispositivos, baseada em algoritmos de ML, são as apresentadas por Kumar, Goyal e Varma (2017), Warden e Situnayake (2020) e Sliwa, Piatkowski e Wietfeld (2020), com a utilização da Artifficial Intelligence of Things (AIoT). Contudo, os modelos de ML implantados nos dispositivos são fixos exigindo a instalação de novos firmwares nos dispositivos quando houver um novo modelo. Essa atitude exige o descomissionamento de todos os dispositivos IoT em uma arquitetura, causando dois graves problemas: os dispositivos param de executar suas funções e a mobilização de profissionais para essa atualização pode se tornar inviável no caso do uso de milhares de dispositivos por ambiente inteligente. Por isso, atualizar o modelo de ML existente no dispositivo IoT sem retirá-lo do seu local de operação é fundamental.

Portanto, embora existam trabalhos que viabilizem o uso de ML em dispositivos IoT ultra restritos, a utilização de FL em dispositivos IoT ultra-restritos usados em redes IoT baseadas em sensores e atuadores, e o aumento da eficiência energética em arquiteturas de FL ainda são lacunas de pesquisa relevantes. Além disso, é importante caracterizar os trabalhos que atuam com FL usando a terminologia “ultra-restrito”. Os trabalhos de Saha, Misra e Deb (2021) e Feraudo et al. (2020), embora apresentem utilizar dispositivos “ultra-restritos”, na verdade utilizam dispositivos robustos, como um SBC, capazes de

executar treinamento dentro do próprio dispositivo - e usam o protocolo IEEE 802.11 como meio de transmissão. Somado a isso, Imteaj et al. (2021) apresenta que, dispositivos IoT que não tenham capacidade suficiente para realizar adequadamente o treinamento de modelos de ML, podem acrescentar heterogeneidade estatística em arquiteturas de FL ao realizar um treinamento deficiente.

Outro fator relevante é a questão do consumo de energia em toda a arquitetura que une a IoT e a FL, conforme se observa em Yang et al. (2020) e Du et al. (2020). Diante disso, a arquitetura de FL aplicada às redes IoT baseadas em sensores e atuadores requer eficiência energética, para que os dispositivos ultra-restritos possam desempenhar suas atividades, pois normalmente são implantados em locais sem assistência humana por longos períodos de tempo (PORTILLA et al., 2019).

Além disso, também é importante avaliar a presença de medições anômalas nas medições oriundas de dispositivos participantes de redes IoT baseadas em sensores e atuadores, pois tais medições podem levar ao aumento dos erros na tomada de decisão (FERRAZ JUNIOR et al., 2019).

Por isso, observa-se lacuna no estado-da-arte com relação à aplicação de FL em redes IoT baseadas em sensores e atuadores para que os dispositivos tenham inteligência para realizarem a tomada de decisão de forma autônoma.

Nesse sentido, o problema de pesquisa identificado é se os dispositivos IoT ultra-restritos podem ser utilizados em uma arquitetura de FL, de maneira a: 1) prover eficiência energética para que os dispositivos IoT ultra-restritos mantenham sua característica da vida útil das baterias durar dias ou meses, frente às constantes trocas de modelos de ML globais recebidos e às constantes realizações de inferência, 2) fornecer anonimidade para os dispositivos IoT ultra-restritos, para que a orquestração e provisionamento seja somente realizada de maneira isolada do núcleo da nuvem, 3) fornecer menor tempo para tomada de decisão, 4) fornecer a tomada de decisão mais acurada frente às novas medições coletadas pelos participantes em virtude da atualização do modelo de ML global a partir dos treinamentos locais de cada participante, e, 4) identificar a presença de medições anômalas nos participantes, uma vez que esses dispositivos podem enviar medições anômalas.

1.2 Hipótese

Considerando os problemas de eficiência energética, anonimidade, menor tempo para a tomada de decisão, tomada de decisão mais acurada frente às novas medições constantemente coletadas, e, a necessidade de identificar medições anômalas, a hipótese apresentada nesta Tese se refere à aplicação da FL para gerar modelos de ML globais que possam ser utilizados por dispositivos IoT ultra-restritos implantados em redes IoT baseadas em sensores e atuadores. Busca-se observar se os dispositivos IoT ultra-restritos podem ser utilizados em uma arquitetura de FL, avaliando se o consumo de energia das constantes trocas de modelo de ML global e realizações de inferência reduzem a vida útil das baterias e inviabilizam o uso desses dispositivos.

A partir dessa hipótese, a proposta contempla fornecer eficiência energética para os dispositivos IoT ultra-restritos, ao propor o treinamento dos modelos de ML locais em dispositivos restritos na camada *Edge* e a fase de testes ou inferência (a tomada de decisão) nos dispositivos IoT ultra-restritos. Nesse sentido, o dispositivo ultra-restrito realiza a tomada de decisão, depois de receber o modelo de ML global transmitido pelo gerenciador, em vez de enviar comandos para o núcleo da nuvem e aguardar por comandos.

A proposta apresentada também fornece privacidade e anonimidade aos dispositivos dos participantes (além da privacidade dos dados que é parte integrante de arquiteturas FL). Isso porque a camada *Edge*, diferentemente das arquiteturas tradicionais de FL, gerencia os dispositivos. Por esse motivo, a camada *Edge* é a responsável por provisionar novas aplicações IoT - e por consequência novos modelos de ML globais no gerenciador. Além disso, os modelos de ML globais são avaliados pelos participantes, que por meio da seleção de variáveis podem substituir os modelos globais existentes quando da avaliação pelos demais participantes.

1.3 Objetivos

O principal objetivo deste trabalho é viabilizar inteligência para que aplicações IoT em cenários de missão crítica (como locais sem assistência humana), tomem decisões baseadas nos dados coletados pelos sensores de dispositivos IoT ultra-restritos e acionem de maneira autônoma seus atuadores. Nesse sentido é necessário projetar e avaliar um *framework* para a utilização de modelos de ML globais gerados por uma arquitetura de

FL em dispositivos IoT ultra-restritos, viabilizando a tomada de decisão inteligente e autônoma no dispositivo IoT ultra-restrito pautado em um modelo de ML global gerado por uma arquitetura de FL.

Dessa maneira, para alcançar o objetivo geral, apresentam-se os seguintes objetivos específicos:

- Fornecer eficiência energética para os dispositivos IoT ultra-restritos para manter o perfil de vida útil das baterias.
- Detectar e avaliar o consumo de energia do sistema de sinalização de mensagens para tráfego seguro fim-a-fim de modelos de ML entre a camada *Edge* e os dispositivos IoT ultra-restritos.
- Avaliar o consumo de energia da camada *Edge Intelligence*, a qual provê privacidade e anonimidade para os dispositivos IoT participantes do *framework* proposto.
- Avaliar o consumo de energia de diferentes modelos de ML em dispositivos IoT ultra-restritos.
- Detectar a presença de medições anômalas em participantes com a conclusão do treinamento dos modelos de ML locais.
- Avaliar o consumo de energia nas camadas *Edge* e *Extreme Edge* considerando a transmissão de modelos de ML globais e a realização de inferências pelos dispositivos IoT ultra-restritos.

1.4 Método de pesquisa

A seguir apresentam-se as etapas do método de pesquisa:

- Pesquisa bibliográfica e levantamento dos trabalhos mais diretamente relacionados à FL, *Edge Intelligence*, redes IoT baseadas em sensores e atuadores, dispositivos IoT ultra-restritos e a integração dessas camadas para a realização de treinamento federado e inferência;
- Análise dos requisitos funcionais para a construção do *framework* proposto;
- Projeto de construção do *framework* proposto;

- Estruturação da bancada de testes usando dispositivos IoT ultra-restritos físicos (não-simulados/emulados);
- Realização de experimentos para validação do *framework* proposto diante das hipóteses apresentadas;
- Análise quantitativa do consumo de energia requerido para utilização do *framework* proposto;
- Análise quantitativa dos resultados do treinamento federado para identificação de participantes com medições anômalas;
- Apresentação dos resultados para a comunidade científica, por meio da divulgação dos resultados em congressos e periódicos especializados.

1.5 Organização

Esta Tese está organizada da seguinte forma: o Capítulo 2, Referencial teórico, apresenta os fundamentos referentes à inteligência artificial, aplicação da inteligência artificial nos dispositivos IoT ultra-restritos, FL, *Edge Intelligence* e as redes IoT baseadas em sensores e atuadores. O Capítulo 3, Trabalhos relacionados, traz um panorama da utilização de FL em dispositivos IoT, bem como apresenta o problema de pesquisa. O Capítulo 4, Framework proposto, apresenta o FedSensor, com os elementos que o compõem para atender o objetivo de viabilizar a FL para os dispositivos IoT ultra-restritos, detectando anomalias e fornecendo eficiência energética. O Capítulo 5, Experimentos, apresenta as características dos experimentos realizados para validação do FedSensor, sendo que no Capítulo 6, Resultados e discussão, discutem-se os resultados obtidos após a realização dos experimentos. O Capítulo 7 apresenta a Conclusão e os futuros trabalhos que podem ser conduzidos com a utilização do FedSensor.

2 REFERENCIAL TEÓRICO

Neste capítulo apresentam-se os principais conceitos apresentados neste trabalho. A Seção 2.1 apresenta a IoT e seus requisitos e aplicações. A Seção 2.2 apresenta a importância do papel da nuvem (núcleo), *Edge* e *Extreme Edge* na IoT. Os conceitos fundamentais sobre redes IoT baseadas em sensores e atuadores e os dispositivos IoT que as compõem estão na Seção 2.3, apresentando também a questão das medições anômalas. A Seção 2.4 trata da importante questão energética na integração nuvem-*edge*-dispositivos. A utilização da inteligência na camada *Edge* consta na Seção 2.5. Nesse sentido, a Seção 2.6 apresenta os conceitos fundamentais de Inteligência Artificial e ML, enquanto a Seção 2.7 trata da inteligência artificial aplicada aos dispositivos IoT. Por fim, a Seção 2.8 contém a fundamentação a respeito das arquiteturas de FL.

2.1 IoT

A IoT viabiliza aplicações em diferentes domínios, como saúde inteligente, transportes e a Indústria 5.0, fornecendo serviços de sensoriamento, monitoramento e atuação (AL-FUQAHA et al., 2015; NI; LIN; SHEN, 2019; HABIBZADEH et al., 2018; ASGHARI; RAHMANI; JAVADI, 2019). Para fornecer esses serviços, as aplicações IoT requerem a união dos recursos fornecidos pelos sensores com a escalabilidade e poder de processamento fornecido pela computação em nuvem, em suas diferentes frentes: nuvem (núcleo) e *Edge* (ELAZHARY, 2019; SAMAILA et al., 2018; SHA et al., 2018).

A IoT fornece capilaridade para as aplicações quando se observam os inúmeros dispositivos IoT conectados, que com seus atuadores podem realizar ações nos ambientes nos quais estão implantados (MIORANDI et al., 2012; AL-FUQAHA et al., 2015). Independentemente dos protocolos de transmissão utilizados pelas diferentes aplicações IoT, a conectividade dos dispositivos é um recurso importante, embora observa-se que a conectividade pode sofrer instabilidade e interrupção (MONTORI et al., 2018).

A heterogeneidade de dispositivos e protocolos gera um desafio para a integração das aplicações IoT nessa união entre nuvem (núcleo), *Edge* e dispositivos, pois não há um protocolo único para tráfego fim-a-fim de dados. Portanto, Čolaković e Hadžialić (2018) apresenta que há esforços na academia, indústria e corporações em gerar sistemas

de transmissão de dados em diferentes camadas (aplicação e rede, principalmente) para atender à heterogeneidade. Mesmo assim, Čolaković e Hadžialić (2018) enfatiza que ainda há uma lacuna de um *framework* que faça a união desses diferentes padrões em uma visão única de IoT para atender aos diferentes domínios em ambientes inteligentes.

Isso porque é crescente o número de dispositivos diariamente conectados e que além de gerarem dados a partir das medições coletadas de seus sensores, também requerem o apoio de uma plataforma que consiga orquestrar e provisionar um grande número de dispositivos e seus serviços atrelados (SHELBY; BORMANN, 2011; ČOLAKOVIĆ; HADŽIALIĆ, 2018). Por esse motivo, se faz necessária uma plataforma para gerenciamento desse grande número de dispositivos, apresentada na seção a seguir.

2.2 Nuvem, Edge e Extreme Edge

Considerando a necessidade de orquestração e provisionamento de inúmeros dispositivos e de serviços das aplicações IoT em diferentes domínios, a computação em nuvem vem justamente atender a essa demanda (KOVACS et al., 2016).

O modelo de serviços fornecidos pela computação em nuvem fornece escalabilidade, disponibilidade e pagamento apenas pelos recursos utilizados, beneficiando as diferentes aplicações IoT, sejam elas com a demanda de alto ou baixo tráfego de dados, poder de processamento e armazenamento (AL-FUQAHA et al., 2015).

O núcleo da nuvem é comumente associado ao alto poder de processamento e capacidade de armazenamento, contudo, também associado a problemas de atraso (*delay*) e latência (AL-FUQAHA et al., 2015; ČOLAKOVIĆ; HADŽIALIĆ, 2018). Nesse sentido, a *Edge Computing* (ou apenas *Edge*) é uma extensão do núcleo da nuvem que está mais próxima dos dispositivos que compõem as aplicações IoT, e por isso oferecem menor latência e atraso.

O crescimento contínuo das complexas operações realizadas na *Edge* faz com que os equipamentos dessa camada tenham cada vez mais poder de processamento, além de serem heterogêneos e suportarem múltiplos protocolos, fato que viabiliza a tomada de decisões apoiada pela IA em diferentes aplicações IoT (PORTILLA et al., 2019).

A camada *Edge* é uma plataforma próxima à fonte de dados que integra recursos de rede, processamento e armazenamento, entre outros, funcionando no mesmo domínio de

utilização dos dispositivos por ela gerenciados. Comparado a um modelo de computação em nuvem, que tem por objetivo a centralização dos dados em uma plataforma robusta, mas que está a muitos saltos de distância dos equipamentos finais, a *Edge* processa os dados no mesmo local de implantação dos dispositivos, reduzindo a latência, o volume de dados transmitidos e o tempo de resposta, ao mesmo tempo que aumenta a segurança e a privacidade, segundo Lu et al. (2020).

A *Edge* conta com a vantagem de estar diretamente ligada aos dispositivos IoT que possuem poder computacional suficiente para executar tarefas complexas. Além disso, a Edge pode ser composta por uma outra camada, a Extreme Edge (PORTILLA et al., 2019), que está na base da IoT e é a responsável por coletar medições do ambiente. Essa coleta de medições na Extreme Edge é realizada por dispositivos compostos por sensores, microcontroladores, um módulo de rádio e bateria. Esses dispositivos podem se comunicar por redes de curta (RSSF) ou longa (LPWAN) distância (PORTILLA et al., 2019).

Diferentemente do propósito original de RSSF e LPWAN (em que os sensores apenas enviam suas medições para um gateway), a camada *Edge* aliada à *Extreme Edge* está um passo além disso: a colaboração entre dispositivos heterogêneos com relação aos protocolos de transmissão de dados (tanto de camada de enlace, quanto de aplicação) permitem a geração de informações valiosas sobre os dados diretamente *on-site* (PORTILLA et al., 2019).

Por isso, a seguir apresentam-se as redes IoT baseadas em sensores e atuadores.

2.3 Redes IoT baseadas em sensores e atuadores

As redes IoT baseadas em sensores e atuadores, localizadas na camada *Extreme Edge*, são compostas por dispositivos que têm por finalidade o sensoriamento e atuação no ambiente em que estão implantados (NASSER et al., 2017).

2.3.1 Dispositivos IoT

Não há uma unanimidade em delimitar os tipos de dispositivos IoT existentes. Isso é comprovado quando são observados os trabalhos de Guha Roy et al. (2018), Khaled et al. (2018), Yazici, Basurra e Gaber (2018), Khaled e Helal (2019), Kim et al. (2019), e Gómez-Carmona et al. (2019) que utilizam como dispositivo IoT um SBC Raspberry

Pi[1], que tem como características 1GB RAM, 1.2 GHz quad-core CPU, saída de vídeo, saída de áudio, entre outros recursos. Outros trabalhos, como Dunkels et al. (2007), Raza, Wallgren e Voigt (2013), Margi, Alves e Sepulveda (2017), Hahm et al. (2016), Zikria et al. (2018), Raza et al. (2017), Vilajosana et al. (2014), e Martinez et al. (2015) se utilizam de dispositivos ultra-restritos em redes LoWPAN e LPWAN.

Nesta Tese, os dispositivos são classificados com base na capacidade de processamento e memória em robustos, restritos ou ultra-restritos.

Dispositivos robustos não tem restrições de processamento, memória, armazenamento, requerem muita energia, e precisam estar diretamente conectados a uma fonte ininterrupta de energia.

Dispositivos restritos, embora não tenham o mesmo poder computacional de dispositivos robustos, suportam sistemas operacionais de propósito geral (como Linux, Windows). Costumam ser utilizados em aplicações de visão computacional, por exemplo, mas requerem muita energia e também precisam estar ligados a uma fonte contínua de energia (como câmeras inteligentes) ou serem recarregados diariamente (como smartphones).

Dispositivos ultra-restritos, em oposto, são aqueles utilizados em RSSF e LPWAN e têm severas restrições de processamento, memória, energia e capacidade de transmissão de dados (normalmente com MTU de até 256 Bytes). A vantagem é que dispositivos ultra-restritos podem ser implantados em sites sem assistência humana e podem operar por meses ou anos sem trocar a bateria, como aponta (MORIN et al., 2017).

Os dispositivos ultra restritos, segundo Hahm et al. (2016) são segregados em 3 classes:

1. Classe 0 (menos de 10KB de RAM e menos de 100KB de ROM/flash);
2. Classe 1 (aproximadamente 10KB de RAM e 100KB de ROM/flash);
3. Classe 2 (aproximadamente 50KB de RAM e 250KB de ROM/flash).

As características dos dispositivos IoT ultra-restritos, portanto, permitem aumentar a longevidade do dispositivo, minimizando a substituição ou recarregamento das baterias (MORIN et al., 2017). O consumo de energia desses dispositivos é muito menor que

[1] https://www.raspberrypi.org/products/raspberry-pi-3-model-b/

o consumo dos restritos, conforme se observa em (FERRAZ JUNIOR et al., 2022) - que durante a comparação considera a execução da mesma atividade: sensoriamento e envio de medição de sensores usando mensagens com segurança fim-a-fim entre um dispositivo e um servidor na camada *Edge*. Para realizarem suas tarefas, as redes IoT baseadas em sensores têm dois componentes fundamentais: o sistema de mensagens e os dispositivos IoT que as compõem.

2.3.2 Sistema de mensagens para redes IoT baseadas em sensores

A conexão de um dispositivo ultra restrito à camada *Edge* depende do tipo de comunicação a ser utilizado: LPWAN (longa distância, normalmente quilômetros) ou LoWPAN (curta distâcia, entre 1 e 100 metros). Redes LPWAN compreendem a utilização de protocolos como Long Range (LoRa) ou SigFox (AL-TURJMAN; MALEKLOO, 2019). A comunicação fim-a-fim com a plataforma Edge ocorre a partir do gateway dessas redes, que recebe a mensagem dos seus dispositivos. Por outro lado, o padrão LoWPAN permite que os dispositivos utilizem o IPv6, gerando assim redes 6LoWPAN. Os dispositivos 6LoWPAN são ultra-restritos e podem entregar mensagens fim-a-fim à camada Edge, utilizando na camada de aplicação mensagens "publish-subscribe".

Segundo Uslu, Okay e Dursun (2020), Al-Masri et al. (2020), Glaroudis, Iossifides e Chatzimisios (2020), Araujo et al. (2019), Chaudhary, Peddoju e Kadarla (2017) e Ferraz Junior et al. (2022), os principais protocolos para transmissão de mensagens publish-subscribe são: eXtensible Messaging and Presence Protocol (XMPP), Advanced Message Queuing Protocol (AMQP), Data Distribution Service (DDS) e Message Queue Telemetry Transport (MQTT). Contudo, Kim et al. (2019) lembra que esses sistemas de mensagens não são adequados para a descoberta dos sensores disponíveis em um dispositivo IoT automaticamente, e por isso a plataforma *Edge* tem papel fundamental na orquestração e provisionamento de equipamentos e seus sensores quando se utilizam desses sistemas de mensagens.

Sistemas de mensagens *publish-subscribe* contemplam a interação entre diferentes equipamentos: o Broker, o *publisher* (publicador) e o *subscriber* (subscritor) (KHALED et al., 2018; KIM et al., 2019). A interação entre os clientes é moderada pelo Broker, portanto, os clientes não precisam se conhecer para trocar mensagens (KIM et al., 2019). Um cliente é uma aplicação ou um dispositivo que troca mensagens em um determinado tópico com

outro cliente. A carga útil é dependente do tópico e é um campo aberto (não há um padrão a seguir para a transmissão dos dados). Contudo, como apresentado por Fischer, Kumper e Tonjes (2019) e Bellavista et al. (2019) é comum o uso de JavaScript Object Notation (JSON) em mensagens *publish-subscribe*. Os tópicos fornecem uma identificação da carga útil MQTT enviada (a que se refere a mensagem).

Conforme apresentado em Ferraz Junior et al. (2022) (resultados preliminares desta Tese), independentemente da carga útil transmitida, os protocolos que geram mensagens com maior número de bytes estão assim ordenados de modo decrescente: XMPP, AMQP, DDS e MQTT. Há vantagens, portanto, na utilização do MQTT para o tráfego de mensagens para interconexão dos dispositivos aos servidores na camada *Edge*.

A *Edge* contém uma plataforma com vários serviços para atender às várias demandas existentes em aplicações, sejam elas IoT ou não. Nesse sentido, há segregação de serviços (em máquinas virtuais ou contêineres) para banco de dados, armazenamento e monitoramento, entre outros. O serviço responsável por receber as mensagens dos dispositivos IoT é o IoT Agent, o qual tem como característica ser específico para tratar mensagens de um determinado protocolo (NAKAGAWA; SHIMOJO, 2017; ARAUJO et al., 2019). Por sua vez, o IoT Broker, contempla um grupo de IoT Agents, sendo o ponto único de contato com os demais serviços sensíveis ao contexto da plataforma em nuvem (KOVACS et al., 2016).

Diante do apresentado, observa-se a importância de um sistema de mensagens para as redes IoT baseadas em sensores, pois assim os dados coletados pelos dispositivos são adequadamente recebidos e tratados pela Edge. As redes IoT baseadas em sensores utilizam diferentes tipos de dispositivos dependendo da necessidade de cada aplicação IoT.

O consumo de energia, contudo, não deve ser um assunto exclusivamente observado pelos dispositivos IoT que compõem uma arquitetura IoT. Por isso, a seção a seguir apresenta como alcançar eficiência energética em toda a cadeia interconectada de uma arquitetura IoT (nuvem-edge-dispositivo).

2.4 A necessidade da eficiência energética

A eficiência energética deve ser observada em qualquer ambiente inteligente. Os trabalhos de Yigitcanlar et al. (2019) e Silva, Khan e Han (2018) apresentam que as cidades

inteligentes precisam ser sustentáveis em primeiro lugar, antes de serem inteligentes.

Nessa seara da sustentabilidade, o trabalho de Riekstin et al. (2018) apresenta uma pesquisa abrangente sobre métricas, métodos e ferramentas para avaliação do consumo de energia em ambientes em nuvem centralizados e distribuídos.

Além disso, os trabalhos de Riekstin et al. (2018), Jalali et al. (2016), e Martinez et al. (2015) apresentam duas formas para modelar o consumo de energia: (1) baseado em tempo, ou (2) baseado em capacidade. Os modelos de consumo de energia baseados em tempo consideram que dispositivos ou equipamentos são compartilhados com poucos usuários - caso dos dispositivos IoT ultra-restritos. Em contrapartida, os modelos baseados em capacidade são os adotados por equipamentos na nuvem e *edge*, pois são altamente compartilhados entre vários usuários. Dessa maneira, para avaliar a eficiência energética, de uma maneira que contemple a integração nuvem-edge-dispositivo, é necessário combinar os dois modelos de consumo de energia.

Segundo Ahvar, Orgerie e Lebre (2022), a proliferação de novas aplicações IoT requer arquiteturas nas quais o processamento ocorre na borda da rede - como é o caso das arquiteturas de FL. Ahvar, Orgerie e Lebre (2022) também exorta que ainda há lacunas sobre o consumo de energia gerado pelo processamento na *edge*.

Mesmo nesse cenário de importância da eficiência energética, observa-se a necessidade de inteligência na camada *Edge*, independentemente do tipo de dispositivo utilizado, conforme se observa na seção a seguir.

2.5 Edge Intelligence

Zhou et al. (2019) atesta que a Edge Intelligence deve ter um entendimento mais amplo do que apenas permitir o treinamento e inferência baseada na inteligência artificial somente nos dispositivos. Por isso, é necessária uma hierarquia colaborativa entre nuvem-edge-dispositivo. Nesse sentido, Zhou et al. (2019) considera seis níveis de Edge Intelligence a seguir apresentados.

O nível 1 (*Cloud-edge Co-inference*) não garante a privacidade dos dados dos dispositivos IoT, portanto não é um modelo elegível para a arquitetura federada, que tem como premissa a privacidade.

O nível 2 (*In-Edge Co-inference and Cloud Training*) define o treinamento exclusivo

na nuvem, não se aproveitando da capacidade de processamento da camada Edge.

O nível 3 (*On-Device Inference and Cloud Training*) requer que a nuvem conheça todos os dispositivos participantes, mas com a geração de um modelo único não atualizável para os dispositivos – nesse cenário a atualização dos modelos depende de dados disponibilizados de forma pública, que podem não fornecer a realidade das medições do ambiente inteligente, e por isso os modelos gerados não têm a acurácia necessária para a aplicação IoT.

O nível 4 (*Cloud-Edge Co-training and Inference*) define o treinamento compartilhado entre edge-cloud, com a inferência podendo ocorrer na nuvem ou na edge – e não no dispositivo.

O nível 5 prevê que o treinamento do modelo e a inferência ocorram na camada Edge. O nível 6 define que o dispositivo realize o treinamento e a inferência. Os níveis 5 (*All In-Edge*) e 6 (*All On-Device*) não se aproveitam da colaboração que a nuvem pode fornecer na geração de modelos com mais acurácia, gerados, por exemplo, com a expansão e utilização de parâmetros utilizados em aplicações IoT similares que estejam em locais diferentes.

Zhou et al. (2019) deixa claro que não há um "melhor nível" e que isso depende da aplicação, mas que há poucos trabalhos que atuam com dispositivos IoT ultra-restritos participantes de arquiteturas de *Edge Intelligence*. Considerando a necessidade de inteligência na *Edge*, a seguir são apresentados os principais conceitos referentes à inteligência artificial.

2.6 Inteligência Artificial e ML

A inteligência artificial tem como propósito o desenvolvimento de entidades, ou agentes, inteligentes (RUSSELL et al., 2016). Um agente inteligente deve ter a capacidade de aprender a partir de exemplos (dados), sendo três os tipos de aprendizado: *supervisionado*, *não-supervisionado* e *por reforço*.

A aprendizagem de máquina encontra no aprendizado estatístico um vasto conjunto de ferramentas para o entendimento sobre os dados, abrangendo os aprendizados supervisionado e não-supervisionado (JAMES et al., 2013).

Segundo James et al. (2013) o aprendizado supervisionado envolve a construção

de modelos estatísticos para previsão baseado em variáveis dependentes de uma ou mais variáveis independentes. No aprendizado não supervisionado há variáveis independentes, mas não se tem as variáveis dependentes ou variáveis resposta. Dessa forna, não há relação de dependência, o que normalmente leva à construção de grupos por similaridade.

No aprendizado por reforço, o agente aprende a tomar decisões interagindo com o ambiente - e com isso recebendo recompensas de acordo com as ações realizadas (SUTTON; BARTO, 1998).

Para que os modelos possam ser aplicados, duas fases importantes devem ser executadas: (1) a de treinamento; e (2) a de teste (esta também denominada inferência). A fase de treinamento envolve o uso de uma base de dados para que o modelo possa prever resultados na fase de testes - no caso do aprendizado supervisionado e não-supervisionado. A fase de testes requer menos poder computacional quando comparada à de treinamento, segundo Gopinath et al. (2019).

Para a realização das fases de treinamento e testes, diferentes modelos de ML podem ser utilizados para atingir os objetivos requeridos pela aplicação, dentre eles *OneClassSVM, Elliptic Envelope, Isolation Forest, Nearest Neighbors, Support Vector Machine (SVM), Neural Networks, Naive Bayes* (CAMINHA; PERKUSICH; PERKUSICH, 2018). O estudo aprofundado de modelos de ML, como por exemplo, avaliação de acurácia, precisão, sensibilidade, sensitividade e outros parâmetros específicos de ML, estão fora do escopo deste trabalho, e por isso não são abordados nesta Tese.

Independentemente do modelo, a ML requer adaptações para que as complexas operações dos algoritmos possam ser realizadas por dispositivos com capacidades reduzidas de processamento, memória e energia. Nesse sentido dá-se origem à inteligência artificial aplicada aos dispositivos, apresentada na seção a seguir.

2.7 Artificial Intelligence of Things (AIoT)

Warden e Situnayake (2020) apresentam que ML é uma técnica usada por equipamentos para prever ações, a qual é baseada em observações anteriores, o que a torna diferente da programação tradicional, em que um software tradicional toma decisões baseadas em regras estabelecidas em rotinas ou tarefas previamente programadas dependendo da entrada do sistema. Decisões baseadas em regras pré-definidas tem sua utilidade,

por exemplo, quando uma temperatura ultrapassa um determinado limiar e um alarme precisa ser disparado. Contudo, no caso de decisões mais complexas que consideram um conjunto de variáveis, pode ser difícil saber a combinação exata de fatores que preveem um determinado resultado.

Para a tomada dessas decisões mais complexas, algoritmos de ML, que resultam em um modelo baseado nos dados fornecidos por meio do processo de treinamento, podem ser utilizados.

Inicialmente os dispositivos IoT foram denominados dispositivos inteligentes - mesmo não tendo nenhuma inteligência, apenas a capacidade de transmitir os dados coletados pelos seus sensores. A incorporação de inteligência artificial nesses dispositivos efetivamente os tornou inteligentes, termo denominado Artificial Intelligence of Things (AIoT) (GUDUR; BALAJI; PEREPU, 2020). A AIoT é necessária pois aplicações IoT sensíveis ao atraso se tornam inviáveis caso tenham que aguardar o recebimento de comandos para tomar uma decisão. Isso se evidencia no trabalho de ajuste de postura apresentado por Yao et al. (2018). Uma das limitações do trabalho é a latência de 4,5 segundos para o envio das medições de acelerômetro para a plataforma em nuvem que resulta em atrasos. Além disso, caso um determinado objeto inteligente precise aguardar um comando do ente externo (da plataforma em nuvem, por exemplo) e houver interrupção da comunicação, o objeto inteligente não receberá o comando e não executará a ação desejada. Em contrapartida a isso, na AIoT, os dispositivos IoT realizam a tomada de decisão baseada em resultados obtidos a partir da execução de modelos de ML. A fase de testes do modelo de ML, que culminará na decisão a ser tomada, é realizada no próprio dispositivo IoT.

Há várias propostas de utilização de AIoT em dispositivos ultra-restritos, como pode se observar em Warden e Situnayake (2020), Sliwa, Piatkowski e Wietfeld (2020), Kumar, Goyal e Varma (2017) e Shalaginov, Semeniuta e Alazab (2019). Para dispositivos robustos, a AIoT é composta pelos softwares padrão executados comumente em desktops e smartphones. Contudo, os dispositivos ultra-restritos não têm capacidade para realizar o treinamento, e por isso é necessário o treinamento em um ente externo, como na computação em nuvem. Ainda assim, a atualização do modelo de ML no dispositivo ultra-restrito ainda é uma lacuna a ser pesquisada.

Para a geração do modelo de ML (seja para aplicações tradicionais apresentadas no Capíulo 1 ou nos modelos implantados em AIoT), o treinamento dos modelos de ML é realizado em um servidor que precisa estar de posse dos dados, resultando na perda da privacidade dessas informações. Além disso, um servidor central que realize o treinamento é obrigado a receber as medições de todos os dispositivos participantes de uma arquitetura IoT fato que gera um alto tráfego de dados (SATTLER et al., 2020).

Para suplantar essas dificuldades, o treinamento distribuído de um modelo de ML entre os participantes de uma mesma arquitetura se faz necessário. Esta necessidade é justamente uma característica encontrada em ambientes de FL.

2.8 Aprendizagem federada

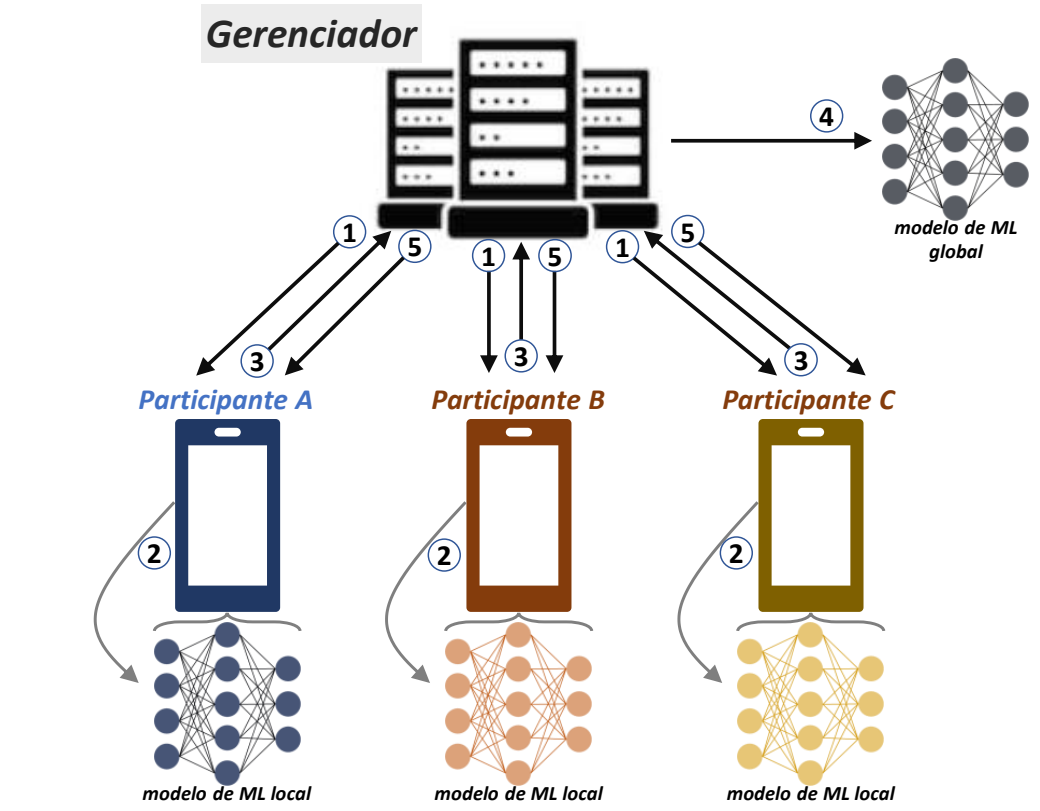
A FL, apresentada inicialmente por McMahan et al. (2017), viabiliza um aprendizado colaborativo de modelo de ML em redes sem fio, na qual os recursos de comunicação (como largura de banda e energia) são limitados, além do objetivo principal, que se refere à privacidade dos dados dos clientes e participantes da arquitetura, conforme apresenta (ZHOU et al., 2020).

2.8.1 Componentes

Em uma arquitetura de FL os principais componentes são: os participantes (clientes), o gerenciador (servidor) e um *framework* de comunicação e consolidação de modelos de ML, este utilizado como meio para realização do treinamento (MCMAHAN et al., 2017). A Figura 1 mostra os componentes de uma arquitetura de FL.

Os participantes são os proprietários dos dados que podem ser, por exemplo, organizações ou dispositivos móveis (QIN; Ye Li; YE, 2021). Em uma arquitetura que contempla clientes com características muito díspares, a arquitetura deve contemplar uma forma de treinamento que não exija a participação de todos os clientes em todas as iterações necessárias para se gerar um novo modelo. Isso porque o uso constante de um dispositivo móvel como participante na geração de novos modelos pode fazer com que a bateria do dispositivo tenha a vida útil reduzida ou ainda, que o dispositivo não responda no tempo devido, resultando em lentidão ou impossibilidade de realização do treinamento. Por isso, em arquiteturas de FL define-se uma fração de clientes a participar de cada iteração do treinamento, segundo Nishio e Yonetani (2019).

Figura 1 – Visão geral de uma arquitetura de FL



Fonte: Adaptado de Qin, Ye Li e Ye (2021) e Qi et al. (2021)

O gerenciador é um servidor com capacidade suficiente para realizar o treinamento de um modelo de ML, em geral, localizado em uma plataforma de nuvem. A finalidade do gerenciador é conduzir o treinamento para geração de um modelo global a ser utilizado por todos os participantes, controlando a comunicação e o número de iterações necessárias para que se atinja a acurácia desejada (MCMAHAN et al., 2017).

O *framework* de comunicação e consolidação de modelos de ML está presente no gerenciador e nos participantes, viabilizando a entrega dos modelos locais gerados nos participantes para o gerenciador (é indicado pelas setas numeradas na 1). Uma vez com os modelos locais recebidos, o gerenciador passa a agregar os modelos locais gerando um novo modelo global. O modelo global, então, é enviado para todos os participantes, dando

início a uma nova iteração.

Esses componentes permitem a uma arquitetura de FL reduzir:

- O risco de acesso não autorizado aos dados (pois os dados não são transmitidos entre o participante e o gerenciador);
- O tráfego de rede, pois o modelo de ML resultante do treinamento é menor que o volume de dados que o modelo representa;
- O tempo e custo da transferência de informações, ao se reduzir o volume total de dados transmitidos;
- Os requisitos de poder computacional do servidor central, bem como os requisitos de armazenamento - pois os dados não estão de posse do gerenciador.

Nesse sentido, utilizar a FL viabiliza novas aplicações, a seguir apresentadas.

2.8.2 Aplicações de FL

Alinhada à garantia de privacidade dos dados durante o treinamento, a FL permite a geração de um único modelo de ML global que considera os dados de diferentes participantes, viabilizando um aprendizado com dados nunca coletados por um participante específico (dados que não constam na base de treinamento de um participante). Essas características favorecem, dentre outras, as aplicações IoT apresentadas a seguir.

Na *saúde inteligente*, os dados hospedados por hospitais e clínicas médicas são sensíveis e devem ser protegidos por quem os hospeda (LI et al., 2020). Nesse sentido, a utilização de um modelo de ML global para a previsão de doenças é uma das soluções para quebrar a barreira de análise de dados de diferentes hospitais. O trabalho de Lee et al. (2018), utiliza a FL para detectar pacientes similares em diferentes hospitais - sem compartilhar os dados dos pacientes. A avaliação de similaridade permite aos médicos extrair características comuns desses diferentes pacientes, indicando o tratamento com mais precisão. Outra aplicação se refere à utilização da FL para gerar um modelo de ML global que, ao avaliar os registros médicos de um paciente, pode prever quando esse paciente com doença cardíaca deve ser hospitalizado. Essa previsão pode ser baseada nas medições fornecidas por dispositivos que medem as condições de saúde do paciente, com

os hospitais armazenando essas informações, e, sem compartilhá-las, poderem gerar um modelo de ML global único com mais precisão na previsão.

Na *Indústria 5.0*, os principais pilares são a sustentabilidade, a inteligência artificial e a inteligência na Edge, segundo Fraga-Lamas, Lopes e Fernández-Caramés (2021). Considerando o volume de dados gerados, estes são sensíveis e o compartilhamento desses dados não é uma opção entre diferentes organizações. A coleta de dados dos sensores das máquinas permite determinar futuros desgastes e falhas com base na utilização de modelos de ML nos próprios equipamentos (XU et al., 2021; MADDIKUNTA et al., 2022). Dessa maneira, os dados provenientes de uma população maior de máquinas aumentam a acurácia do modelo de ML global para manutenção preditiva (HAFEEZ; XU; MCARDLE, 2021). Todas as aplicações na Indústria 5.0, contudo, não podem negligenciar a sustentabilidade. Por isso, a questão de eficiência energética é um recurso obrigatório a ser observado e que deve ser avaliado em qualquer cenário que faça a união de inteligência artificial na camada *Edge* usando dispositivos IoT ultra restritos, segundo (FRAGA-LAMAS; LOPES; FERNÁNDEZ-CARAMÉS, 2021).

Os Intelligent Transportation Systems (ITS) também se beneficiam da arquitetura de FL. Veículos com equipamentos com alto poder de processamento e conectividade 5G (como é o caso dos veículos da marca Tesla podem ser participantes de arquiteturas de FL, fazendo o treinamento do modelo de ML local dentro do próprio veículo (POSNER et al., 2021). O próprio trabalho de Posner et al. (2021) apresenta uma Federated Vehicular Cloud (FVC), na qual um veículo é o gerenciador (gerando o modelo de ML global), enquanto os demais são considerados participantes (gerando os respectivos modelos de ML locais);

No domínio das cidades inteligentes, o trabalho de Huang et al. (2021) apresenta o FedParking, um *framework* de FL permitindo a diferentes organizações operadoras de estacionamentos realizarem a previsão de espaço disponível nos estacionamentos de forma privada - esses dados são sensíveis, pois contém informações como horário de entrada e saída, bem como a taxa de ocupação e o valor cobrado. Os veículos são parte integrante da arquitetura, pois o veículo submete ao FedParking uma solicitação de vaga, que é indicada pelo *framework* (HUANG et al., 2021).

Ainda no domínio das cidades inteligentes, o *Índice de Qualidade do Ar (IQAr)* é

uma aplicação frequentemente avaliada, conforme se observa em (CHHIKARA et al., 2021). Com o aumento da população nas cidades, têm-se constatado o aumento da poluição do ar, uma vez que que muitos dos poluentes identificados no meio ambiente resultam de ações humanas. Observa-se como principal culpado o Particulate Matter (PM) 2.5 (CHHIKARA et al., 2021). Para medir a severidade da poluição atmosférica, é usado o IQAr, que é determinado ao se analisar a concentração de vários materiais particulados, como o PM 2.5, monóxido de carbono (CO), ozônio (O_3), entre outros. Nesse sentido, o trabalho de Chhikara et al. (2021) apresenta uma arquitetura de FL para previsão de IQAr utilizando *drones* para a coleta de medições e treinamento do modelo local.

Dentre todas as aplicações apresentadas, observa-se uma característica comum: o aprendizado colaborativo de um modelo de ML.

2.8.3 Aprendizado colaborativo

Na abordagem apresentada por McMahan et al. (2017), há K dispositivos (*smartphones*, no caso), denominados participantes. Os participantes recebem um modelo global $w_{t_0}^G$ de um servidor central e passam a gerar modelos locais segundo a Eq. 2.1).

$$w_t^k \leftarrow w - \gamma \nabla \mathcal{L}(w; \mathcal{D}_k) \tag{2.1}$$

Na Eq. 2.1 considera-se a função de perda definida no modelo global previamente recebido $\mathcal{L}(\mathbf{w}; x)$, bem como seu conjunto de dados local $\mathcal{D}_k$ e uma taxa de aprendizado γ. O servidor central recebe modelos locais dos participantes e realiza uma agregação (no caso de McMahan et al. (2017), com o uso do `FedAvg`, realiza-se a média dos modelos recebidos conforme a Eq. 2.2).

$$\mathbf{w}_{t+1}^G \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} \mathbf{w}_{t+1}^G \tag{2.2}$$

Gera-se, com o uso da Eq. 2.2, um novo modelo global $\mathbf{w}_{t+1}^G$ - que é repassado para os dispositivos. Essas iterações ocorrem até que se atinja a acurácia necessária, ou que se tenha executado o número de iterações estipulado.

A FL, portanto, se baseia em quatro passos segundo Qin, Ye Li e Ye (2021):

1. Atualização local de modelo: cada cliente atualiza seu conjunto de dados $\mathcal{D}_k$, permitindo assim gerar novos modelos locais de acordo com os dados obtidos de seus respectivos sensores;

2. Envio de parâmetros/coeficientes: cada cliente envia seus coeficientes, ou seja, seu modelo local ($\mathbf{w}_t^k$) treinado para o gerenciador;

3. Agregação global: o servidor central calcula a média dos coeficientes recebidos dos participantes, gerando um novo modelo global ($\mathbf{w}_{t+1}^G$);

4. *Feedback*: o servidor central envia os novos coeficientes (o novo modelo de ML global) para os participantes, para a próxima iteração.

Observa-se que a transmissão de modelos entre o gerenciador e o participante está no centro de uma arquitetura de FL, recurso analisado a seguir.

2.8.4 Transmissão de modelos entre participante e gerenciador

Os participantes de uma arquitetura de FL precisam transmitir e receber modelos do gerenciador, exigindo conectividade constante com alto volume de tráfego de dados (SATTLER et al., 2020).

O tráfego no gerenciador, embora intenso ao distribuir o modelo de ML global para todos os participantes (naquela iteração), ainda é reduzido quando comparado ao recebimento de medições de todos os equipamentos para um treinamento local - isso sem considerar a questão da privacidade dos dados, que seria quebrada. Contudo, enquanto o servidor central está normalmente na nuvem (que tem largura de banda suficiente para um alto volume de transmissão de dados), não se pode exigir o mesmo dos participantes, segundo Sattler et al. (2020). Os participantes são heterogêneos, pois podem ser smartphones, SBC e servidores de porte intermediário na *Edge*, entre outros. Ao se considerar a utilização de participantes com hardware muito diferentes, o gerenciador precisa conhecer as características de hardware de todos os participantes, para escolher adequadamente os participantes de cada iteração para geração do modelo global. Por isso, o tempo de treinamento do modelo de ML local é diferente para cada participante.

Segundo Sattler et al. (2020), o número total de bits que precisam ser transferidos (download/upload) de cada participante durante o treinamento é apresentado na

equação 2.3:

$$b^{up/down} \in \mathcal{O}(N_{iter} \times f \times |\mathcal{W}| \times (H(\Delta \mathcal{W}^{up/down}) + \eta)) \tag{2.3}$$

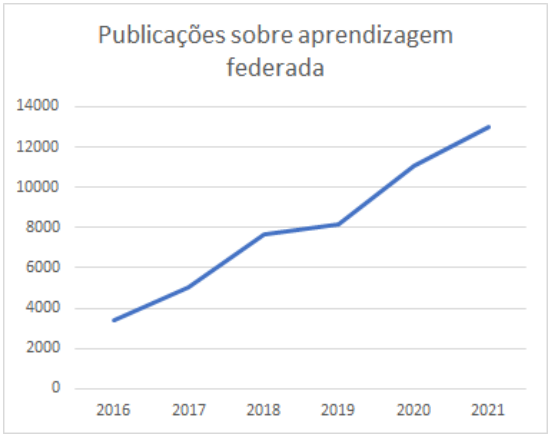
onde N_{iter} é o número total de iterações realizadas pelos participantes, f é a frequência de comunicação em que um treinamento é solicitado pelo gerenciador, $|\mathcal{W}|$ é o tamanho de bits do modelo de ML, $H(\Delta \mathcal{W}^{up/down})$ é a entropia das atualizações dos parâmetros do modelo durante as ações de *upload* e *download*, e η é a ineficiência de precisão na transmissão do modelo entre gerenciador e participante.

Portanto, para que a geração do modelo de ML global seja realizada com sucesso, a arquitetura de FL gera um tráfego muito intenso para os dispositivos, podendo causar, dentre outros, dois problemas principais: (1) um aumento do consumo de energia do dispositivo, que desestimula a participação dos dispositivos na arquitetura de FL, e; (2) alta latência para atualização do modelo global, uma vez que o servidor central deve aguardar a resposta de todos os participantes. Para superar essa última dificuldade, o servidor central pode descartar modelos locais recebidos de um participante que demorou para enviar seu modelo local. Contudo, segundo Nishio e Yonetani (2019), isso gera alguns problemas, como o consumo desnecessário de energia do participante, e a perda de parâmetros/coeficientes que poderiam aprimorar o modelo global. Nesse sentido observa-se a importância do apoio da Edge Intelligence na FL.

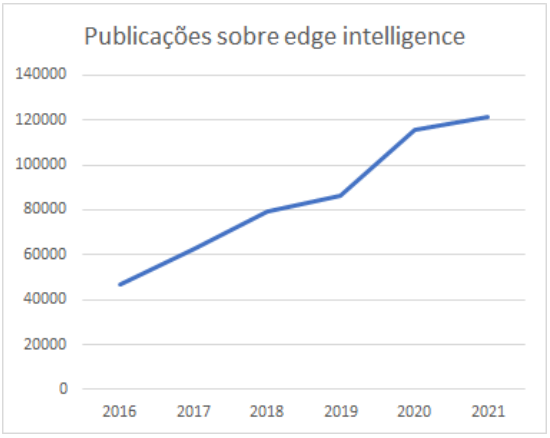
3 TRABALHOS RELACIONADOS

Neste capítulo analisam-se as propostas existentes na literatura a respeito do treinamento colaborativo de modelos de ML usando FL, *Edge Intelligence* e redes IoT. A intenção é observar como os trabalhos fornecem a privacidade dos dados e dispositivos, bem como avaliam a eficiência energética.

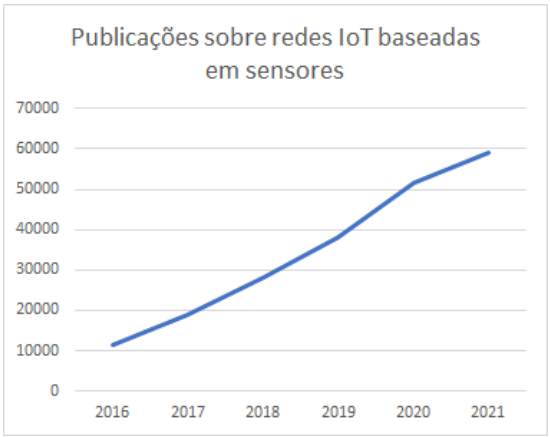
Figura 2 – Número de publicações levantadas (desde 2016) sobre os principais temas abordados nesta Tese



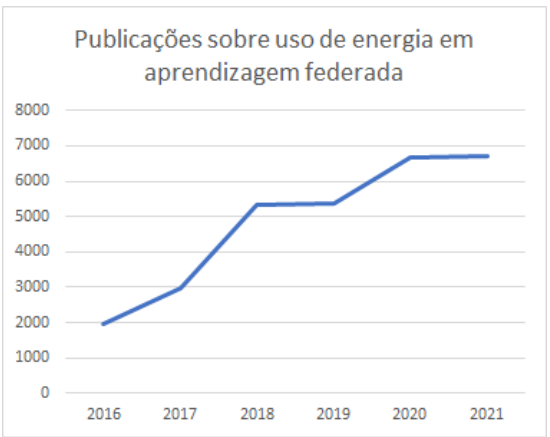
(a) Aprendizagem federada



(b) Edge Intelligence



(c) Redes IoT baseadas em sensores



(d) Uso de energia em aprendizagem federada

Fonte: Autor

De acordo com as informações coletadas no *website* dimmensions.ai[1], conforme mostra a Figura 2, o número de publicações referentes a FL (Figura 2a), *Edge Intelligence*

[1] plataforma comercial para pesquisa de publicações científicas https://dimmensions.ai/discover/publication. Acessado em: 15/06/2022.

(Figura 2b), redes IoT baseadas em sensores (Figura 2c), e consumo de energia em FL (Figura 2d) têm aumentado de maneira significativa desde 2016 (as chaves de pesquisa utilizadas são, respectivamente, *federated learning*, *edge intelligence*, *sensor-based IoT networks*, *federated learning energy*).

Nesse sentido, considerando os temas de pesquisa apresentados na Figura 2, os trabalhos a seguir apresentados, e que estão mais diretamente relacionados ao escopo desta Tese, contemplam: FL, transmissão de modelos de ML globais ou locais para dispositivos e necessidade de uso de energia em FL.

Os trabalhos de Abdulrahman et al. (2021), Saha, Misra e Deb (2021) e Feraudo et al. (2020) apresentam que a união da FL com a utilização de dispositivos IoT ultra-restritos se mostra um desafio devido aos seguintes fatores: (1) ao volume de dados referentes à transmissão de parâmetros, (2) à capacidade computacional incompatível com os recursos desses dispositivos, e (3) à heterogeneidade estatística.

O trabalho de McMahan et al. (2017) é considerado o primeiro trabalho a definir e demonstrar a aplicação da FL utilizando dispositivos IoT - no caso smartphones. Apresenta-se o `FedAvg`, que realiza a agregação média dos modelos de ML locais, gerando o modelo de ML global. Contudo, o trabalho de McMahan et al. (2017) não fornece privacidade para os dispositivos (fornece apenas para os dados), pois requer acesso aos dispositivos para transmitir e receber os modelos de ML, bem como não contempla o apoio da camada *Edge* no treinamento.

Por sua vez, Saha, Misra e Deb (2021) apresenta um mecanismo de escolha de um gerenciador a cada vez que um modelo de ML global é gerado. A proposta se baseia na escolha de um gerenciador dentre o conjunto de nós Fog de uma arquitetura. A intenção é escolher o nó de Fog que esteja mais próximo dos outros nós cujos dispositivos serão utilizados nas iterações de geração do modelo de ML. Contudo, mesmo que o gerenciador seja escolhido considerando a distância para os outros nós, ainda é necessário que esse gerenciador conheça e troque dados com todos os dispositivos participantes que realizarão o treinamento do modelo de ML local.

O trabalho de Ye et al. (2020) requer que os dispositivos realizem o treinamento das camadas mais baixas de uma Convolutional Neural Network (CNN). Esse conceito é inviável para os dispositivos IoT ultra-restritos, pois estes não são capazes de executar

nem as camadas mais baixas de treinamento de uma rede neural, sendo que o treinamento precisa ocorrer fora do dispositivo, de acordo com o apresentado por Zhou et al. (2019).

Outro trabalho relacionado, o de Foukalas e Tziouvaras (2021) apresenta a importância de um protocolo para troca de mensagens em arquiteturas de FL, baseado no Constrained Application Protocol (CoAP), bem como exige o treinamento local no dispositivo, o que se mostra inviável para dispositivos IoT ultra-restritos. No mesmo sentido, Feraudo et al. (2020) apresenta um protocolo para transmissão de modelos entre a nuvem e dispositivos IoT baseado no MQTT. A proposta é utilizar o Broker como mediador da comunicação entre gerenciador e participantes, embora exija que os dispositivos realizem o treinamento do modelo de ML local.

Outro viés comparativo nos trabalhos relacionados é a questão energética, abordada por Peng et al. (2021). Contudo, Peng et al. (2021) apenas avalia a energia para a seleção dos melhores participantes para a geração do modelo global. Por sua vez, Ren et al. (2019) aborda o consumo de energia dos dispositivos (embora use apenas dispositivos robustos e restritos), mas considera que os dispositivos executam o treinamento *on-device* (o que é inviável para dispositivos ultra-restritos), bem como define que os dispositivos obtém energia do Edge Server, o que não se mostra uma realidade nos cenários IoT com dispositivos espalhados sem assistência humana. Além disso, o trabalho também apresenta que a energia usada para o processamento local nos dispositivos IoT é constante, o que não se comprova com o uso de dispositivos IoT ultra-restritos, pois o volume de dados trafegados e o tempo de uso de CPU influenciam no consumo de energia, conforme se observa nos resultados preliminares desta Tese, publicados em Ferraz Junior et al. (2021a), e Ferraz Junior et al. (2022).

Por todo o exposto, observa-se que os dispositivos IoT ultra restritos precisam do apoio da Edge Computing para poderem participar de uma arquitetura de FL. Dentro da revisão de literatura pesquisada e dos trabalhos relacionados, não são identificados trabalhos que façam a união da FL com dispositivos IoT ultra-restritos.

As propostas que se utilizam da Fog ou Edge Intelligence aplicando FL - caso dos trabalhos de Saha, Misra e Deb (2021), Wang et al. (2020), Ye et al. (2020) - exigem um mínimo de treinamento realizado no dispositivo. Isso resulta na perda de privacidade do dispositivo, pois o gerenciador - que realiza o treinamento do modelo global - precisa

conhecer todos os dispositivos participantes, estejam esses dispositivos dentro do próprio ambiente inteligente ou não. Por exemplo, em uma arquitetura de FL de saúde inteligente, dispositivos com conexão 5G de diferentes hospitais responderiam a uma mesma estação rádio-base. Essa estação rádio-base, portanto, conheceria todos os dispositivos em seu alcance, resultando na perda de privacidade desses dispositivos - mesmo que os dados nunca sejam transmitidos, apenas os modelos locais.

Aliada à questão do treinamento, a eficiência energética também é fundamental nessa união nuvem-edge-dispositivos, tanto com relação ao consumo de energia para os dispositivos IoT realizarem a inferência e a tomada de decisão usando modelos de ML, quanto em relação ao consumo de energia da nuvem e do edge nesse cenário de cooperação entre as camadas para a realização do treinamento. Um dispositivo IoT ultra-restrito não pode ter um modelo de ML que consuma uma quantidade de energia que o deixe operando por poucos dias, uma vez que dispositivos ultra-restritos devem operar por meses ou anos, conforme aponta Morin et al. (2017). Diante disso, em qualquer proposta de FL que contemple dispositivos IoT ultra-restritos, a avaliação do consumo de energia é um requisito obrigatório, conforme apresentado por Ferraz Junior et al. (2019), Ferraz Junior et al. (2021a), Elsts et al. (2020).

Tabela 1 – Comparação com os principais trabalhos relacionados na literatura sobre aprendizagem federada em redes IoT

Trabalho	**Arquitetura**	**Avaliação de energia**	**Tipos de dispositivos**	**Privacidade (dados) e Anonimidade (dispositivos)**
McMahan et al. (2017)	Nuvem-dispositivos	Não	Robustos	apenas dos dados
Saha, Misra e Deb (2021)	*Fog*-dispositivos	Sim	Restritos	apenas dos dados
Ye et al. (2020)	Nuvem-*Edge*-dispositivos	Não	Restritos	apenas dos dados
Foukalas e Tziouvaras (2021)	Nuvem-*Edge*-dispositivos	Não	Restritos	apenas dos dados
Feraudo et al. (2020)	Nuvem-dispositivos	Não	Restritos	apenas dos dados
Peng et al. (2021)	Nuvem-dispositivos	Sim	Restritos	apenas dos dados
Ren et al. (2019)	*Edge*-dispositivos	Sim	Restritos	apenas dos dados
Shalaginov, Semeniuta e Alazab (2019)	Nuvem-dispositivos	Não	Ultra-restritos	sem privacidade

Com a utilização de dispositivos IoT ultra-restritos, uma questão adicional comumente abordada é a anomalia nas medições. Conforme se observa em Al-Amri et al. (2021), Ferraz Junior et al. (2019) e Raza et al. (2017), os dispositivos IoT ultra-restritos são sus-

cetíveis a anomalias na coleta das medições dos seus sensores. Portanto, uma proposta que utilize dispositivos IoT ultra-restritos para a tomada de decisão deve considerar a detecção de anomalias nos dados fornecidos pelos dispositivos para a realização do treinamento.

Considerando esse escopo, apresentam-se na Tabela 1 os trabalhos mais diretamente relacionados ao apresentado nesta Tese.

Ao observar o panorama apresentado na Tabela 1, constata-se que não há trabalhos integrando uma arquitetura de FL com a *Edge Intelligence* usando dispositivos ultra-restritos participantes de redes IoT baseadas em sensores. O único trabalho que utiliza dispositivos ultra-restritos, o de Shalaginov, Semeniuta e Alazab (2019) apenas transmite um modelo de ML para o dispositivo, mas não garante a privacidade dos dados, nem dos dispositivos, pois o treinamento requer os dados brutos coletados pelos dispositivos.

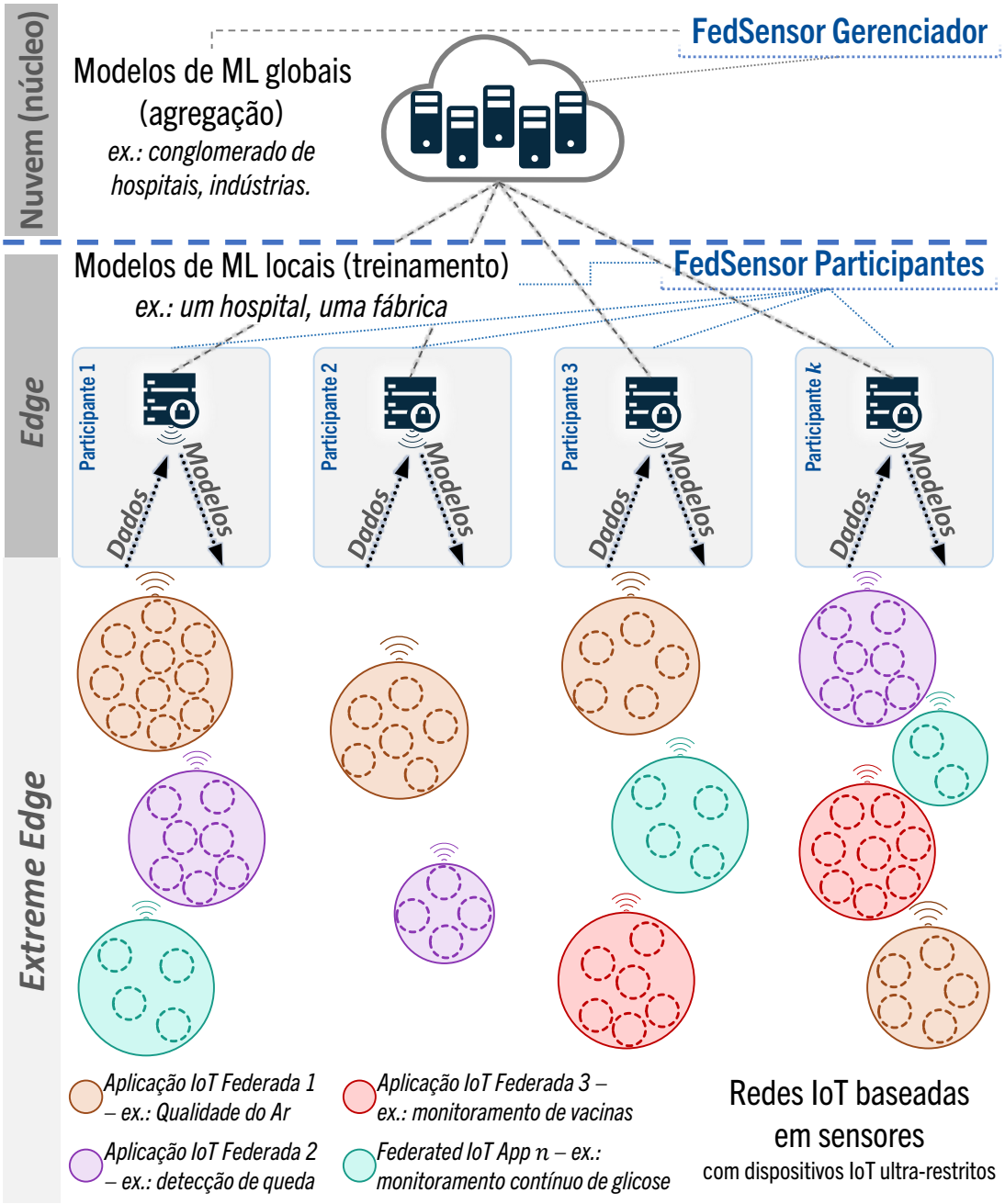
4 FRAMEWORK PROPOSTO

Para atingir os objetivos, este trabalho apresenta o **Federated learning for Sensor-based IoT networks framework (FedSensor)**, que utiliza a FL para gerar os modelos de ML globais em um servidor central na nuvem, e transmitir esses modelos para os dispositivos IoT ultra restritos por meio da Edge Intelligence (que atua como participante gerando os modelos de ML locais). Dessa maneira, o dispositivo IoT ultra-restrito é capaz de participar da arquitetura de FL realizando a inferência após receber o modelo de ML global.

O FedSensor mantém as características de privacidade dos dados, pois a *Edge* está no mesmo local ou região do dispositivo IoT ultra-restrito. A Figura 3 apresenta a visão geral da proposta deste trabalho. O FedSensor é guiado pelas características dos dispositivos IoT ultra-restritos provisionados em cada participante. Esse provisionamento permite ao FedSensor gerar modelos de ML apropriados para esses dispositivos, bem como se aproveita das características de múltiplos sensores que podem estar disponíveis nesses dispositivos para buscar por modelos com mais acurácia. Portanto, as principais contribuições inéditas deste trabalho são as seguintes:

- *Framework*: o FedSensor gera modelos globais customizados para serem utilizados por dispositivos IoT ultra-restritos implantados em RSSF e LPWAN;
- Federação de Aplicações IoT: Dispositivos IoT contém vários sensores e, mesmo sem estarem fisicamente próximos (nem estarem no mesmo ambiente inteligente), podem ser utilizados para a mesma finalidade. Em cada participante geram-se agrupamentos lógicos de dispositivos que contêm os mesmos sensores, as mesmas características de processamento, memória e energia e são capazes de executar os mesmos modelos de ML. Portanto, como os participantes não se conhecem (garantindo assim a privacidade não somente dos dados, mas também dos dispositivos IoT que geram os dados), diferentes participantes podem provisionar aplicações IoT com modelos de ML iguais (mesmas variáveis preditoras e mesmo desfecho do modelo de ML, como classificação, por exemplo). Por conseguinte, o FedSensor viabiliza a geração de um modelo de ML global único para a mesma aplicação IoT com mais acurácia para

Figura 3 – Visão geral do proposto FedSensor



Fonte: Autor

ser usado pelos participantes e dispositivos IoT, a partir dos modelos de ML locais gerados pelas aplicações IoT federadas em cada participante;

- Modelos de ML locais: o provisionamento dos dispositivos, seus sensores e os modelos de ML suportados, é realizado no participante. Mantendo a privacidade dos dados fornecidos pelos dispositivos IoT sob seu controle, o participante envia metadados

(modelos suportados, sensores disponíveis) para o gerenciador, que por sua vez usa esses metadados para identificar as variáveis preditoras disponíveis e o desfecho do modelo. Assim, o gerenciador pode identificar quando um modelo de ML global com mais acurácia pode ser consolidado na mesma aplicação IoT federada;

- Detecção de anomalias: Dispositivos usados em RSSF e LPWAN podem enviar medições anômalas para o participante, mesmo sendo dispositivos confiáveis (pois os dispositivos podem coletar medições anômalas, oriundas de erro ou defeito em seus sensores) (FERRAZ JUNIOR et al., 2019). Dessa maneira, mesmo um participante confiável (selecionado por meio de Blockchain conforme apresenta Posner et al. (2021), por exemplo) precisa identificar as medições anômalas independentemente do modelo gerado;

- Seleção de variáveis preditoras e desfechos: para gerar modelos de ML com maior acurácia, cada participante observa os dispositivos provisionados para verificar se a ativação de sensores disponíveis (mas não utilizados) nos dispositivos melhoram a acurácia do modelo. Portanto, cada participante gera um novo modelo inicialmente não provisionado para os dispositivos, buscando pelo aumento da acurácia. Ao identificar um modelo com maior acurácia, inicia-se o processo de avaliação do novo modelo junto aos demais participantes para avaliar se o novo modelo tem mais acurácia quando comparado ao modelo de ML global em uso, avaliando com os dados dos demais participantes. Outra vantagem surge, pois a seleção de variáveis e desfechos pode encontrar modelos com acurácia similar mas com diferentes variáveis preditoras e números de desfechos. Dessa maneira, o FedSensor pode fornecer eficiência energética ao escolher um modelo com menos variáveis (e por consequência utilizando menos sensores ou menos classes ou grupos nos desfechos), mas com a mesma acurácia.

O FedSensor utiliza dispositivos IoT ultra-restritos e considera a cooperação entre nuvem e Edge para a geração de modelos de ML, transmitidos com segurança fim-a-fim. A proposta tem como objetivo atender aos problemas identificados no Capítulo 3.

A segurança apresentada nesta Tese contempla a proteção dos modelos de ML trafegados no *framework*, especialmente no tráfego entre o participante e os dispositivos IoT ultra-restritos. Contudo, a maior parcela da contribuição está relacionada à privacidade

dos dados, pois não deixam o ambiente em que estão implantados, uma vez que trata como o dado é coletado, compartilhado e utilizado na geração de modelos de ML globais.

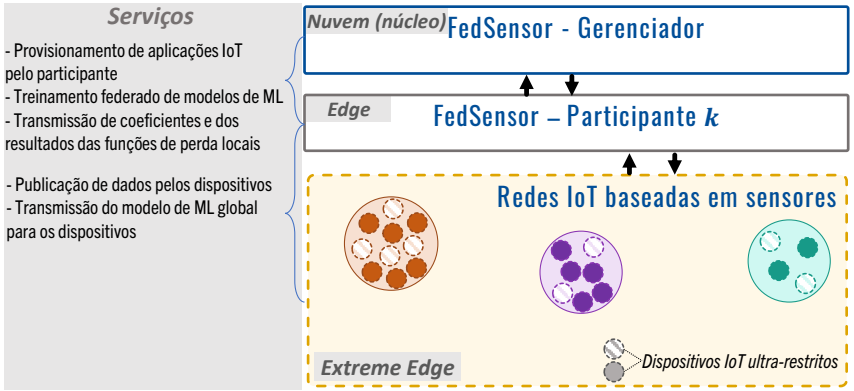
Observa-se que a população que contém todos os dados necessários para as corporações, governos, cidadãos ou comunidade científica cresce muito rapidamente. Consequentemente, o problema de otimização na prática fica improvável ou impossível de se realizar em um único nó, conforme apresentado por Konečný et al. (2016), requerendo a arquitetura distribuída fornecida pelo FedSensor para ajudar a resolver esse problema. A seguir, apresenta-se uma visão geral do FedSensor.

4.1 Visão geral dos elementos fundamentais do FedSensor

No FedSensor, há três elementos principais: o **gerenciador** (na camada *nuvem*), o **participante** (na camada *Edge*), e o **dispositivo IoT ultra-restrito** (na camada *Extreme Edge*).

A Figura 4 apresenta os elementos fundamentais do FedSensor.

Figura 4 – Elementos fundamentais da proposta



Fonte: Autor

O gerenciador, normalmente, é composto por servidores centrais, comumente fornecidos por um provedor de serviços de nuvem, sendo o responsável pela tarefa de agregação dos modelos de ML globais. O participante é um *Edge Server* implantado no ambiente de operação dos dispositivos, sendo normalmente micro estações rádio-base ou *micro clouds* em centros de dados locais - esses Edge Servers estão a apenas um salto de distância dos dispositivos. Os participantes geram os modelos de ML locais de cada aplicação IoT sob seu domínio. Os dados nunca saem da camada *Edge*, mantendo a

privacidade dos dados. Os dispositivos IoT ultra restritos são provisionados e gerenciados pelo participante, garantindo assim a privacidade e anonimidade dos dispositivos, pois o gerenciador não tem acesso aos dispositivos IoT ultra-restritos (não conhece endereços IP, portas, sensores disponíveis em cada dispositivo, fabricante, nem nenhuma outra característica).

Diante do exposto e conforme se observa nos trabalhos de McMahan et al. (2017), Li et al. (2021), He et al. (2020) e Konečný et al. (2016), o principal objetivo da FL consiste em aprimorar o modelo global por meio da minimização da função de perda e do consequente aumento de acurácia. Como o foco deste trabalho está relacionado à utilização e atualização dos modelos de ML (global e local) para os dispositivos localizados na camada *Extreme edge*, este trabalho não abrange as questões estatísticas relativas às variáveis aleatórias serem independentes e identicamente distribuídas (i.i.d.). Observa-se, no trabalho de Sarkar, Narang e Rai (2020), que o problema a ser solucionado pela FL diz respeito ao fato dos dados serem parciais e nem sempre contarem com amostras de todas as situações possíveis, resultando na negativa da hipótese de i.i.d. Contudo, por não ser o escopo deste trabalho, este tema não é tratado nesta proposta. Além disso, a análise de questões específicas de ML como acurácia, precisão, sensitividade, entre outros, também não fazem parte do escopo deste trabalho.

Considerando o escopo apresentado e a visão geral da necessidade de aplicação da FL, apresenta-se a seguir o processo de geração dos modelos de ML no FedSensor.

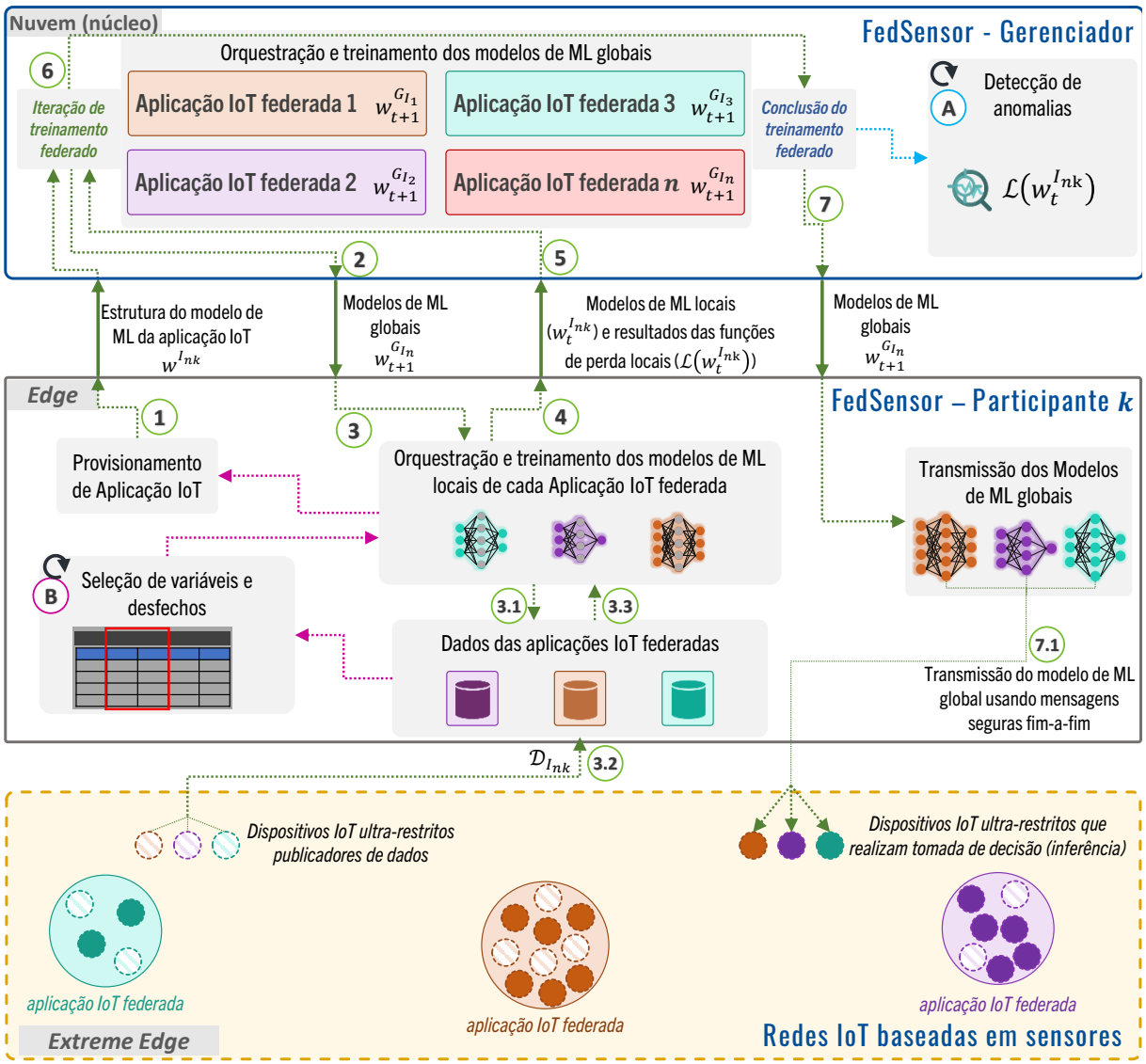
4.1.1 Processo de geração de modelos de ML globais

Para a geração dos modelos, primeiro apresentam-se as características do gerenciador e dos participantes no FedSensor.

O gerenciador é o proprietário dos modelos de ML globais utilizados pelas diversas aplicações IoT federadas. O participante é o proprietário dos dados e dos modelos de ML locais. O dispositivo IoT ultra-restrito, por sua vez, realiza a inferência e a tomada de decisão, depois de receber o modelo de ML global gerado, uma vez que contém atuadores. Adicionalmente, há dispositivos IoT ultra-restritos que participam do FedSensor enviando as medições coletadas, sendo, portanto, os provedores e publicadores dos dados.

A integração entre os componentes do FedSensor e os dispositivos IoT ultra-restritos dentro da arquitetura de FL apoiada pela *Edge Intelligence* consta na Figura 5.

Figura 5 – Componentes do FedSensor *framework*



Fonte: Autor

Como se observa na Figura 5, no FedSensor o provisionamento dos modelos de ML de cada aplicação IoT é executado em cada participante ①, diferentemente das abordagens tradicionais de FL, em que o gerenciador dá início ao processo de aprendizagem. O participante gerencia os próprios dispositivos IoT ultra-restritos, conhecendo os sensores disponíveis em cada dispositivo e os modelos de ML que cada dispositivo pode executar. Este conceito, que é definido neste trabalho como "Federação de Aplicações IoT", apresentado na seção 4.2 a seguir.

Por sua vez, no gerenciador, estão provisionados e disponíveis todos os modelos de ML de cada aplicação IoT federada. O gerenciador envia os modelos de ML globais para

os participantes ②.

Na sequência, ao receber o modelo de ML global do gerenciador, o participante inicia o treinamento do modelo de ML local ③. O participante observa o próprio conjunto de dados (3.1), alimentado pelos dispositivos IoT ultra-restritos publicadores de dados (3.2) e realiza o treinamento local (3.3). Depois de concluída a rodada de treinamento, o participante envia o modelo de ML local e o resultado da função de perda para o gerenciador ④.

O gerenciador recebe todos os modelos de ML locais e os resultados das funções de perda de todos os participantes ⑤ e, com isso, realiza a agregação dos modelos de ML locais de todos os participantes envolvidos naquela rodada de treinamento ⑥. Quando estão concluídas todas as rodadas de uma iteração do treinamento federado, conclui-se o treinamento. Neste momento o gerenciador envia o modelo de ML global final para os participantes ⑦, que por sua vez envia o modelo de ML global para os dispositivos IoT ultra-restritos que realizam a tomada de decisão, por meio da inferência nos dados por ele coletados (7.1).

Além das iterações de treinamento federado, no FedSensor há mais duas etapas considerando as características das redes IoT baseadas em sensores: a detecção de anomalias Ⓐ e a Seleção de variáves/sensores dos dispositivos Ⓑ, etapas apresentadas, respectivamente, nas seções 4.4 e 4.5. Isso porque, conforme apresentado no Capítulo 3, a existência de anomalias nas medições coletadas por dispositivos IoT ultra-restritos é uma característica comum e a detecção de anomalias é um recurso bastante recomendável nessas condições. Por isso, a seleção de variáveis/sensores do dispositivo é um componente obrigatório e que está presente no FedSensor. Adicionalmente, como a eficiência energética é um componente fundamental de qualquer proposta que se utilize de dispositivos IoT ultra-restritos, a análise do consumo de energia é realizada em toda a cadeia interconectada do FedSensor, que compreende "nuvem-Edge-dispositivo IoT ultra-restrito". Avalia-se também o consumo de energia da constante atualização dos modelos de ML globais e realização da inferência nos dispositivos IoT ultra-restritos, com a finalidade de verificar o número de dias de duração da bateria dos dispositivos IoT ultra-restritos com o emprego do FedSensor.

4.2 Federação de Aplicações IoT

As aplicações IoT estão englobadas em domínios verticais, conforme apresenta Al-Fuqaha et al. (2015). Os domínios verticais são ambientes inteligentes, por exemplo:

hospitais e cidades inteligentes, entre outros. Esses domínios verticais são compostos por diferentes aplicações IoT (medição da qualidade do ar de uma cidade, detecção de queda de idosos e monitoramento contínuo da glicose, entre outras). Os ambientes inteligentes baseados no FedSensor contêm um participante, responsável pelas aplicações IoT existentes e pelos modelos de ML que cada aplicação pode utilizar. Posteriormente, as aplicações IoT são provisionadas no gerenciador pelos participantes.

As aplicações IoT consolidam e vinculam as informações sobre os dispositivos IoT, os sensores disponíveis em cada dispositivo, os metadados dos sensores, os modelos de ML suportados, os modelos de ML disponíveis para cada dispositivo e a relação variável-sensor.

Os dispositivos IoT e os sensores deles seguem o padrão de metadados e registro segundo a IPSO Alliance Internet Protocol for Smart Objects (IPSO) Alliance OMA LightweightM2M (LWM2M) (2017), e assim fornecem a padronização necessária para a identificação dos tipos de sensores e as instâncias dos sensores nos dispositivos.

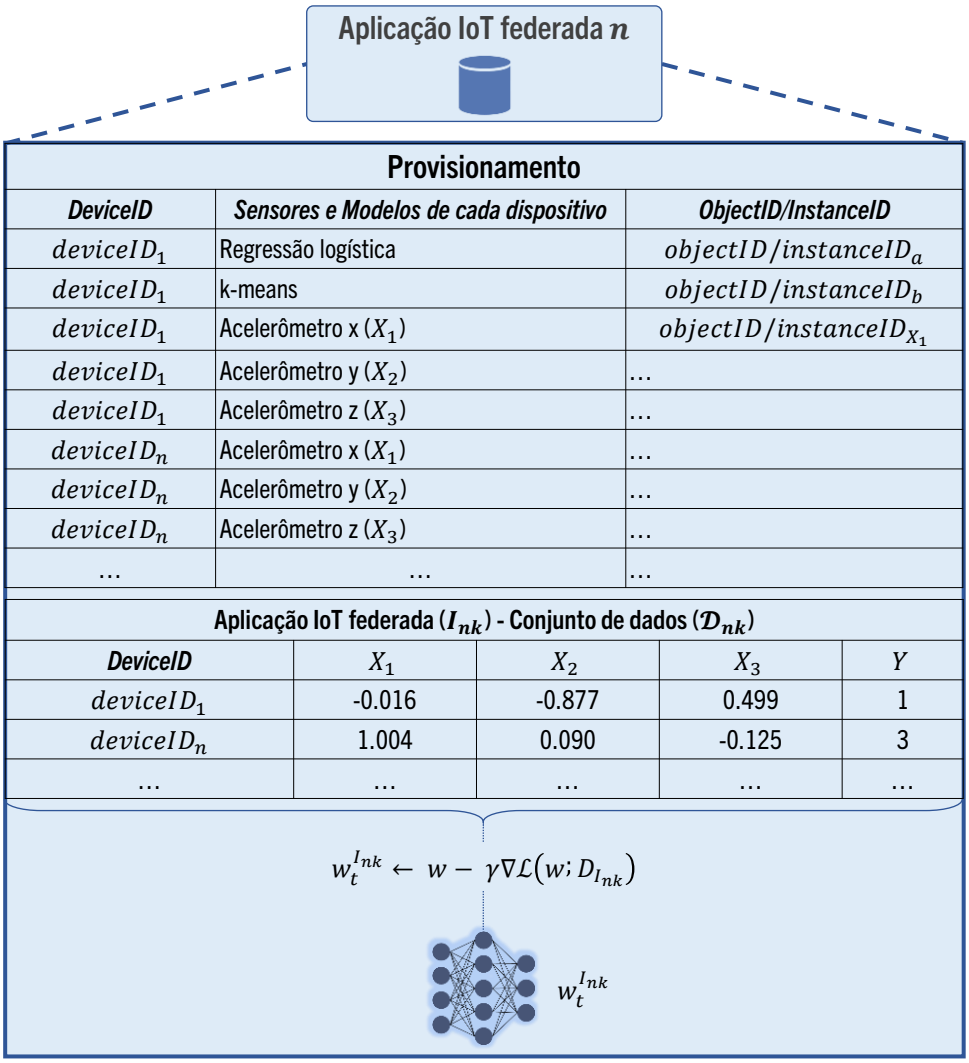
Conforme apresenta a Figura 6, o participante mantém uma base de dados para armazenar os metadados daquela aplicação IoT federada, que contempla a identificação dos dispositivos IoT ultra-restritos que compõem aquela aplicação, os sensores existentes em cada dispositivo e os modelos de ML suportados por aquela aplicação IoT. Os sensores dos dispositivos IoT ultra-restritos são as variáveis do modelo de ML provisionado.

O participante mantém a rastreabilidade dos modelos de ML suportados e em execução em cada dispositivo IoT. O participante, portanto, tem gestão sobre os sensores/atuadores e os modelos de ML suportados pelos dispositivos, para que o modelo de ML possa ser implantado e utilizado pelos dispositivos IoT.

Apresenta-se, na Figura 7, uma visão da integração das diferentes aplicações IoT federadas no FedSensor evidenciando os papéis realizados pelo gerenciador e participantes.

Os dispositivos IoT ultra-restritos são gerenciados pelo participante, usando uma identidade única (denominada *deviceID*) para cada dispositivo, bem como uma identificação única para cada sensor do dispositivo, usando o LightWeight Publish-Subscribe system for ultra-low power IoT devices (LWPubSub), uma contribuição desta Tese conforme se observa nas seguintes publicações: Ferraz Junior et al. (2021a), Ferraz Junior et al. (2021b) e Ferraz Junior et al. (2022). Adiciona-se ao LWPubSub os metadados referentes aos modelos de ML, os quais são provisionados considerando as características de cada

Figura 6 – Estrutura de uma aplicação IoT

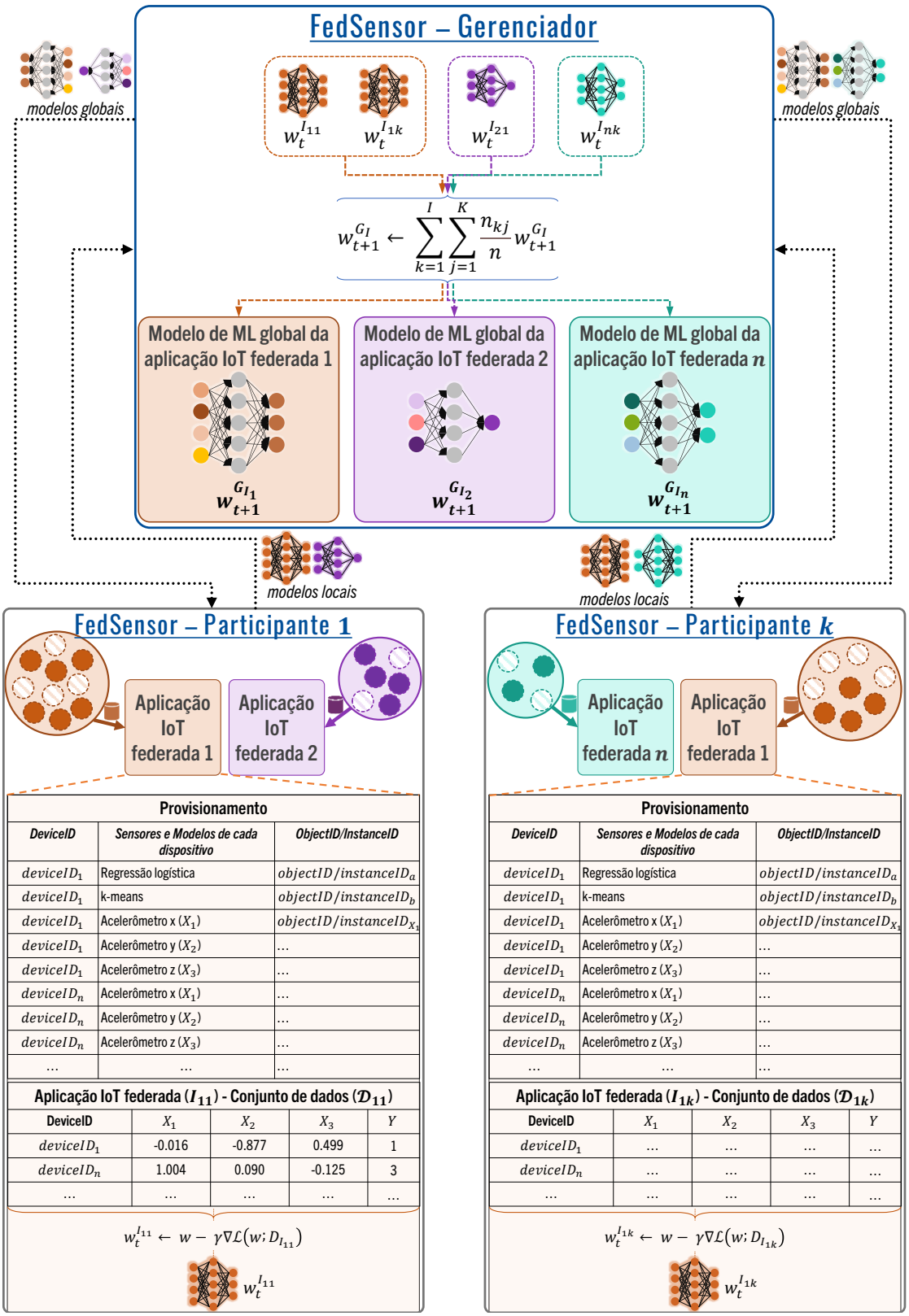


Fonte: Autor

dispositivo IoT ultra-restrito.

Como um exemplo da privacidade nos domínios verticais, uma aplicação IoT pode ser o Índice de Qualidade do Ar em hospitais (estes, tendo em seus *Edge Servers* os participantes). Hospitais que participam da mesma arquitetura de FL, embora não troquem informações entre si sobre sua estrutura interna (dispositivos, sensores e modelos, entre outras), podem ser logicamente agrupados no gerenciador (federação de aplicações IoT), pois podem usar dispositivos com sensores similares para a mesma aplicação IoT (por exemplo, o IQAr). Dessa forma, mesmo os diferentes participantes não se conhecendo, o gerenciador é capaz de agrupar as aplicações IoT com a mesma finalidade, permitindo a geração de modelos globais com o apoio de mais participantes.

Figura 7 – Estrutura das aplicações IoT federadas no gerenciador e nos participantes



Fonte: Autor

No participante, conforme ilustra a Figura 7, o *Edge Server* orquestra e provisiona os próprios dispositivos, gerenciando aqueles que são utilizados para a tomada de decisões e aqueles que enviam medições para subsidiar a geração dos modelos locais e as aplicações IoT do domínio.

Ainda no participante, o conjunto de dados ($\mathcal{D}_{\mathcal{I}_{nk}}$) passa pela detecção de anomalias para posteriormente ser observado pelo agente de seleção de variáveis/sensores dos dispositivos para geração de novos modelos (essas características são tratadas nas próximas subseções).

Uma vez provisionadas as aplicações IoT no gerenciador e a estruturação das aplicações IoT federadas, inicia-se o processo de treinamento cooperativo entre nuvem e *Edge* que resultam nos modelos de ML globais de cada aplicação IoT, apresentados na seção subsequente.

4.3 Treinamento cooperativo entre nuvem e edge

Conforme ilustrado na Figura 7, os dispositivos enviam suas medições ao participante, que por sua vez as armazena no respectivo conjunto de dados da aplicação IoT sob seu controle ($\mathcal{D}_{\mathcal{I}_{nk}}$, onde n indica a aplicação IoT e k a qual Participante os dados da aplicação IoT se referem).

Para a geração do modelo de ML local, cada Participante, para cada aplicação IoT ($\mathcal{I}$), aplica a Eq. 4.1.

$$w_t^{\mathcal{I}_{nk}} \leftarrow w - \gamma \nabla \mathcal{L}(w; \mathcal{D}_{\mathcal{I}_{nk}}) \tag{4.1}$$

onde $w_t^{\mathcal{I}_{nk}}$ é o modelo gerado para cada aplicação IoT, γ é a taxa de aprendizado, e $\mathcal{L}(w; \mathcal{D}_{\mathcal{I}_{nk}})$ é a função de perda a ser minimizada em cada iteração do treinamento local.

As funções de perda diferem de acordo com o modelo de ML aplicado. Apresentam-se exemplos de funções de perda nas Equações 4.2 (erro quadrático médio), 4.3 (entropia cruzada binária), 4.4 (soma dos erros ao quadrado) - usadas respectivamente em modelos de regressão linear, regressão logística e agrupamento com k-means (não são exploradas todas as funções de perda possíveis por não ser escopo deste trabalho).

$$EQM = \frac{1}{N}\sum_{j=1}^{N}(y_i - (\beta_0 + \beta_1 x_{1,j} + ... + \beta_n x_{n,j}))^2 \tag{4.2}$$

$$H_p(q) = -\frac{1}{N}\sum_{i=1}^{N} y_i \; log(p(y_i)) + (1 - y_i) \; log(1 - p(y_i)) \tag{4.3}$$

$$SEQ = \sum_{j=1}^{k}\sum_{i \in C_j} dist(x_i, \mu_j)^2 \tag{4.4}$$

Assim que o modelo de ML das aplicações IoT é $(w_t^{\mathcal{I}})$ minimizado e gerado, o participante o envia para o gerenciador, que por sua vez consolida os modelos de ML das aplicações IoT federadas, gerando assim um único modelo de ML global para cada aplicação IoT. Na Figura 7 observa-se que $\mathcal{I}_{11}$ e $\mathcal{I}_{1k}$ se referem a uma aplicação IoT que se utiliza de dados de acelerômetro, e pode ser utilizada para detecção de queda de idosos, por exemplo. Os Participantes 1 e k não se conhecem, mas se utilizam do mesmo modelo de ML com a mesma finalidade, e enviam seus modelos locais para o gerenciador. O gerenciador, portanto, utiliza os modelos recebidos para gerar um modelo global com mais acurácia, enviando esse novo modelo para os participantes, segundo a Equação 4.5.

$$w_{t+1}^{G_{\mathcal{I}}} \leftarrow \sum_{k=1}^{\mathcal{I}}\sum_{j=1}^{K}\frac{n_{kj}}{n} w_{t+1}^{G_{\mathcal{I}}} \tag{4.5}$$

onde $w_{t+1}^{G_{\mathcal{I}}}$ é o modelo global agregado pelo gerenciador, considerando cada aplicação IoT $\sum_{k=1}^{\mathcal{I}}$ e o respectivo modelo local de cada participante $\sum_{j=1}^{K}\frac{n_{kj}}{n} w_{t+1}^{G_{\mathcal{I}}}$.

Em relação aos dados no FedSensor: cada participante i contém uma matriz $\mathbf{X}_i = [x_{i1}, ..., x_{iK_i}]$ de dados armazenados no conjunto de dados de cada aplicação IoT $(\mathcal{D}_K)$, onde K é o número de medições coletadas pelos dispositivos participantes daquela aplicação IoT, e cada elemento x_{ik} é um vetor de dados utilizado para geração do modelo local. Por sua vez, y_{ik} é o resultado de x_{ik}, em que o vetor $y_i = [y_{i1}, ..., y_{iK_i}]$ é utilizado no treinamento do modelo local. $\mathcal{I}$ representa a aplicação IoT existente em cada participante.

O resultado do treinamento local considerando $\mathbf{X}_i$ e y_i é o vetor $\mathbf{w}_i$, que contém os parâmetros do modelo local ($\mathbf{w}_i$ é o modelo local de um Participante i). Por exemplo, em

uma regressão logística binária, $x_{ik}^T \mathbf{w}_i$ representa a previsão, sendo que $\mathbf{w}_i$ é o vetor de parâmetros que determina o comportamento do algoritmo de regressão linear. Nesse sentido, no FedSensor, para cada participante i, o treinamento local tem por objetivo encontrar os parâmetros que otimizam o modelo (resultando em $\mathbf{w_i^*}$), de maneira a minimizar a função de perda conforme a Eq. 4.6.

$$\underset{\mathbf{w_1},\ldots,\mathbf{w}_\mathcal{I}}{min} \frac{1}{K} \sum_{i=1}^{\mathcal{I}} \sum_{k=1}^{K_j} \mathcal{L}(\mathbf{w}^i; x_{jk}, y_{jk}) \tag{4.6}$$

onde $K = \sum_{i=1}^{I} K_i$ é o tamanho total dos dados de treinamento de uma aplicação IoT, e $\mathcal{L}(\mathbf{w}^i; x_{ik}, y_{ik})$ é a função de perda. A função de perda permite observar o comportamento do modelo de ML frente às atividades de regressão, classificação e agrupamento - de acordo com a função utilizada. A função de perda permite observar o erro na comparação das previsões com os valores verdadeiros. A observação do resultado da função de perda permite observar, a cada rodada do treinamento federado, a tendência do resultado do treinamento. Uma vez que a função de perda apresenta o erro das previsões, o propósito é que esse erro seja reduzido a cada rodada do treinamento federado.

A atualização do modelo $\mathbf{w}_i$ de cada aplicação IoT $\mathcal{I}$ existente em cada participante k depende do modelo global $\mathbf{w}_G$, que por sua vez também depende dos modelos locais. Por exemplo, quando essa iteração, usa o gradiente descendente estocástico, é possível otimizar o modelo global com a geração de $\mathbf{w_G^*}$.

O gerenciador envia o modelo de ML global para os participantes, que por sua vez enviam o modelo para os dispositivos IoT. Os dispositivos IoT ultra-restritos estão inseridos dentro das respectivas redes IoT baseadas em sensores (RSSF ou LPWAN) na camada *Extreme Edge* e, de acordo com as características provisionadas no participante, realizam a inferência, ou o envio dos dados dos sensores para o *Edge Sever*.

Um exemplo fim-a-fim de utilização do FedSensor pode se dar com a detecção de qualidade do ar e deflagração (de maneira inteligente e autônoma pelo dispositivo IoT ultra-restrito) de uma ação como o acionamento de ventilação ou alarme caso a condição do ar esteja perigosa. Inicialmente, o participante provisiona a aplicação IoT no gerenciador, informando: 1) os sensores a serem utilizados nos dispositivos (e seus respectivos objectID/instanceID), 2) os modelos de ML disponíveis nos dispositivos (também no formato objectID/instanceID). O gerenciador, por sua vez, envia o modelo de ML global

para todos os participantes que desejam se utilizar dessa aplicação IoT. Os dispositivos IoT ultra-restritos publicadores de dados passam a alimentar a o conjunto de dados dos seus respectivos participantes, que por sua vez realizam o treinamento do modelo de ML local. Ocorre o treinamento federado (conforme anteriormente apresentado) e, ao final do treinamento, os dispositivos IoT ultra-restritos que realizam a tomada de decisão recebem: 1) os sensores a serem utilizados, 2) o modelo de ML global (e seus coeficientes), 3) o tempo para realização da inferência, 4) o valor máximo aceitável (regressão linear), ou a classe (regressão logística) ou grupo (k-means) que deflagram a ação a ser tomada, e, 5) o atuador a ser acionado caso o resultado da inferência esteja dentro da ação recebida. Exemplos de atuação contemplam acionar: sinal sonoro (alarme), ventilação, luz, entre outros.

Os dados transmitidos pelos dispositivos IoT ultra-restritos viabilizam ao *Edge Server* a geração dos modelos locais. Contudo, conforme apresentado no Capítulo 3, dados desses dispositivos podem conter anomalias. Portanto, antes da utilização dos dados para a geração dos modelos locais, aplica-se a detecção de anomalias, apresentada na seção a seguir.

4.4 Detecção de anomalias

A detecção de anomalias é um recurso bastante recomendável a ser aplicado nas medições obtidas por redes IoT baseadas em sensores (LI et al., 2019).

Em uma arquitetura de FL, a detecção de anomalias tem por objetivo impedir que os modelos gerados forneçam resultados a partir dos quais os dispositivos tomem decisões indevidas. Por exemplo, no caso de uma aplicação IoT que detecta queda de idosos usando como variáveis os dados de acelerômetro (nos eixos x, y, z), uma medição anômala pode fazer com que uma queda não seja identificada, ou que um movimento normal seja caracterizado como queda, fazendo com que o dispositivo IoT tome uma decisão incorreta.

As anomalias emitidas por dispositivos IoT ultra-restritos podem ser oriundas de:

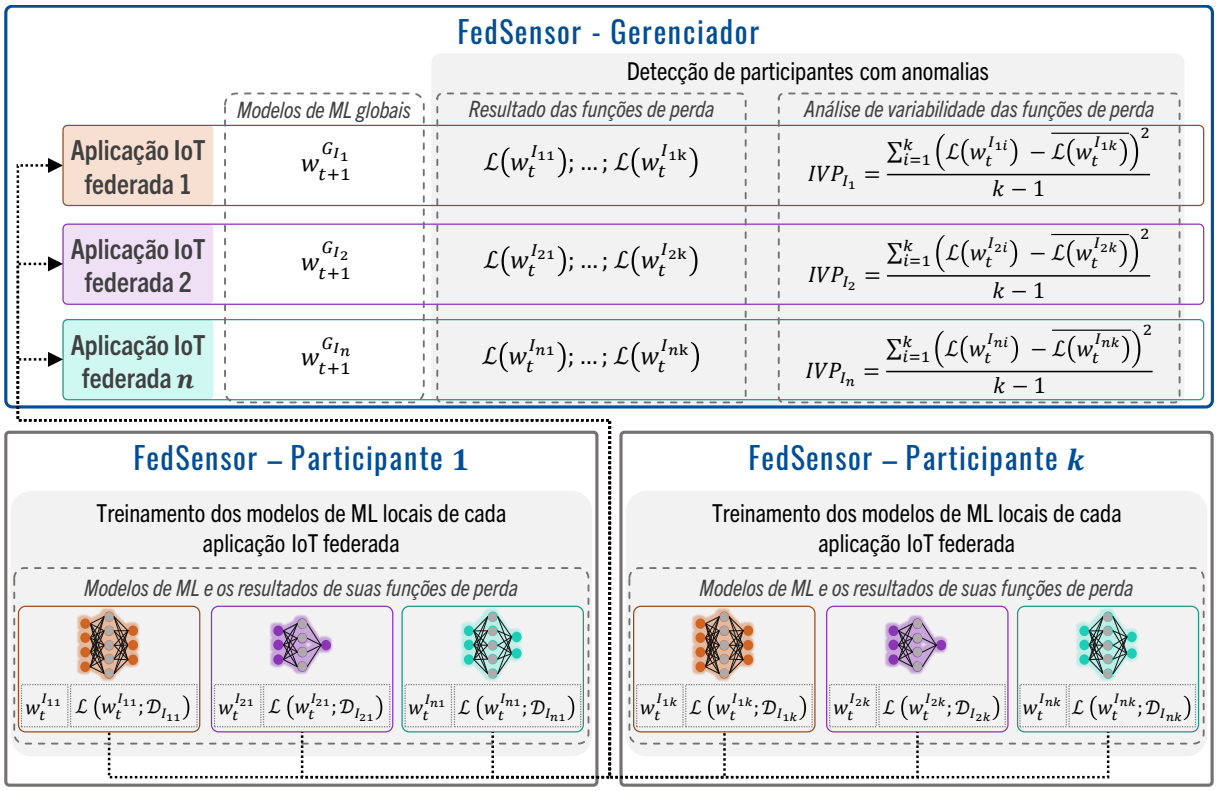
- Medições anômalas por falha nos sensores (LI et al., 2019);
- Medições anômalas por ataques (FERRAZ JUNIOR et al., 2019; NGUYEN et al., 2019; CUI et al., 2018).

Independentemente da origem das medições anômalas, identificá-las é importante para que se reduza o erro (função de perda) durante a realização do treinamento (GARCIA-FONT; GARRIGUES; RIFÀ-POUS, 2016).

Além disso, a identificação de medições anômalas também faz com que os modelos atinjam mais acurácia (COOK; MISIRLI; FAN, 2020; AL-AMRI et al., 2021). As anomalias não costumam fazer parte dos modelos de ML provisionados, pois, em geral, não há ciência sobre quais anormalidades podem ocorrer (REN; ANICIC; RUNKLER, 2021).

A Figura 8 apresenta como o FedSensor atua na identificação de participantes que contém medições anômalas. Nessa Figura, observa-se que os participantes geram os modelos de ML locais das diferentes aplicações IoT federadas ($w_t^{\mathcal{I}_{ni}}$). Uma vez gerados os modelos de ML locais, também são gerados os resultados das funções de perda de cada aplicação IoT federada em cada participante ($\mathcal{L}(w_t^{\mathcal{I}_{ni}})$). Tanto os modelos quanto os resultados das funções de perda são transmitidos para o gerenciador.

Figura 8 – Detecção de participantes com medições anômalas no gerenciador com a análise de variabilidade do resultado das funções de perda



Fonte: Autor

O gerenciador, por sua vez, realiza a agregação dos modelos de ML recebidos

de cada aplicação IoT de cada participante, segundo o apresentado na seção anterior. Posteriormente, conforme apresentado na Figura 8, o gerenciador observa a variabilidade das funções de perda para identificar participantes cujas aplicações IoT tenham anomalias nas medições. A Equação 4.7 apresenta a forma de análise da variabilidade das funções de perda, com a apresentação do Índice de Variabilidade do Participante (IVP).

$$IVP_{\mathcal{I}_n} = \frac{\sum_{i=1}^{k} \left(\mathcal{L}(w_t^{\mathcal{I}_{ni}}) - \overline{\mathcal{L}(w_t^{\mathcal{I}_{nk}})} \right)^2}{k-1} \tag{4.7}$$

Onde $\mathcal{L}(w_t^{\mathcal{I}_{ni}})$ é a função de perda de uma aplicação IoT de um participante, $\overline{\mathcal{L}(w_t^{\mathcal{I}_{nk}})}$ é a média das funções de perda, n é o número de aplicações IoT, k é o número de participantes e $IVP_{\mathcal{I}_n}$ é a variabilidade observada para uma determinada aplicação IoT.

Nesse sentido, valores de funções de perda que reduzem progressivamente resultam em variabilidades baixas - o que é esperado de um treinamento de modelo de ML em aplicações IoT iguais cujos dados estão espalhados em diferentes participantes. Em oposto, valores muito diferentes das funções de perda dos diversos participantes que atuam no FedSensor aumentam o IVP, indicando a presença de anomalias nas medições. Isso porque cada rodada de treinamento federado (dentro da mesma iteração que resultará no modelo de ML global) tem a finalidade de utilizar diferentes dados para treinamento para minimizar o erro e gerar um modelo de ML global com mais acurácia. Esses dados são provenientes dos dispositivos IoT publicadores de dados, apresentados na Figura 5.

Uma vez com o modelo preparado para identificar medições anômalas, aplica-se a seleção de variáveis para realizar a busca pelo melhor modelo para a aplicação IoT.

Considerando que os dispositivos IoT podem ter vários sensores, um agente de seleção de variáveis/sensores dos dispositivos IoT ultra-restritos observa os conjuntos de dados, como proposto na seção a seguir.

4.5 Seleção de variáveis e desfechos

No FedSensor, o provisionamento da aplicação IoT no participante implica em fornecer todas as ***variáveis*** (sensores) disponíveis, bem como diferentes possibilidades de ***desfecho*** da aplicação, como por exemplo: classificação e agrupamento, ou classificação, agrupamento e regressão. Nos diferentes modelos de ML provisionados em cada aplicação

IoT, deve-se fornecer o desfecho: o valor máximo aceitável (regressão), ou a classe (classificação) ou grupo (agrupamento) que deflagrará a ação a ser tomada. Esse provisionamento é realizado por um agente externo, normalmente o detentor dos dados no seu respectivo ambiente inteligente.

A seleção de variáveis e desfechos analisa os conjuntos de dados para gerar novos modelos locais incluindo ou removendo sensores para a coleta de dados. A intenção é buscar os melhores coeficientes para o modelo gerado (variáveis e desfechos). Com isso, o participante submete os novos coeficientes e desfechos gerados no modelo de ML local para o gerenciador, que atualiza o modelo de ML global. A Figura 9 apresenta a seleção de variáveis e desfechos aplicada no FedSensor.

Conforme apresentado na Figura 9, o FedSensor no participante busca por novas variáveis considerando os desfechos provisionados ①. Depois, inicia-se o treinamento federado do novo modelo, envolvendo os demais participantes ②. Avalia-se a acurácia o novo modelo de ML global gerado ($acc_{w_{novo}^{G_{\mathcal{I}_n}}}$) em comparação com a acurácia do modelo de ML global atual ($acc_{w_{atual}^{G_{\mathcal{I}_n}}}$). Caso se encontre uma acurácia superior no modelo novo, o modelo atual é substituído ⑤. Caso a acurácia seja igual, inicia-se o processo de avaliação de eficiência energética, para verificar se o novo modelo de ML global requer menos energia quando comparado ao modelo de ML global atual ④. Caso o novo modelo de ML global exija menos bateria, ele substitui o modelo em uso ⑤.

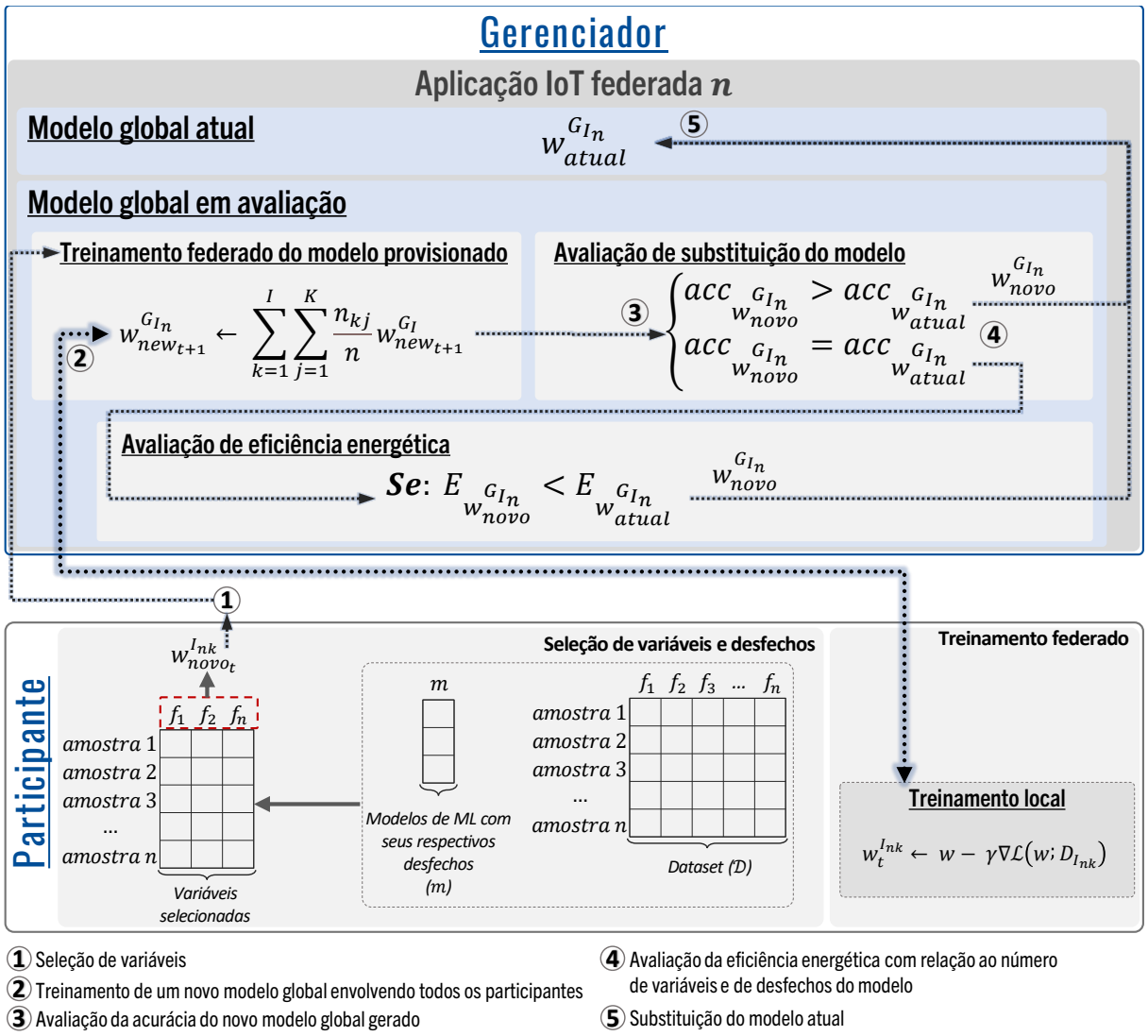
O objetivo é encontrar o conjunto de variáveis e os desfechos que forneçam mais acurácia e economia de energia para o modelo (LIU et al., 2021).

Um exemplo de vantagem de se utilizar a seleção de variáveis e os desfechos é a possibilidade do FedSensor capacitar hospitais com diferentes volumes de dados ($\mathcal{D}_{\mathcal{I}_{nk}}$) a utilizarem o modelo global mais otimizado. Dessa forma, hospitais com poucos dados podem obter o melhor modelo para uma determinada aplicação IoT. Um outro cenário possível pode ocorrer quando um dispositivo coletar dados nunca vistos em um determinado hospital. Se este tipo de dado desconhecido tiver sido contemplado na geração do modelo de ML global, a parir de outro hospital, o dado pode ser entendido e levado em consideração na tomada de decisão.

Aplicado ao FedSensor, a seleção de variáveis e desfechos tem os seguintes objetivos:

1. Identificar as variáveis que fornecem mais acurácia;

Figura 9 – Seleção de variáveis e desfechos aplicada ao FedSensor



Fonte: Autor

2. Identificar (dentro os algoritmos disponíveis para aquela aplicação IoT federada), qual modelo fornece mais acurácia;

3. Identificar quais desfechos fornecem mais acurácia;

4. Reduzir o consumo de energia.

A seleção de variáveis passa por um filtro posterior com o objetivo de reduzir o consumo de energia, avaliando se a redução de variáveis e de desfechos pode contribuir para a redução do consumo de energia, propiciando mais eficiência energética. Conforme

consta na Figura 9, o filtro atua na escolha de menos variáveis e desfechos quando houver modelos com a mesma acurácia.

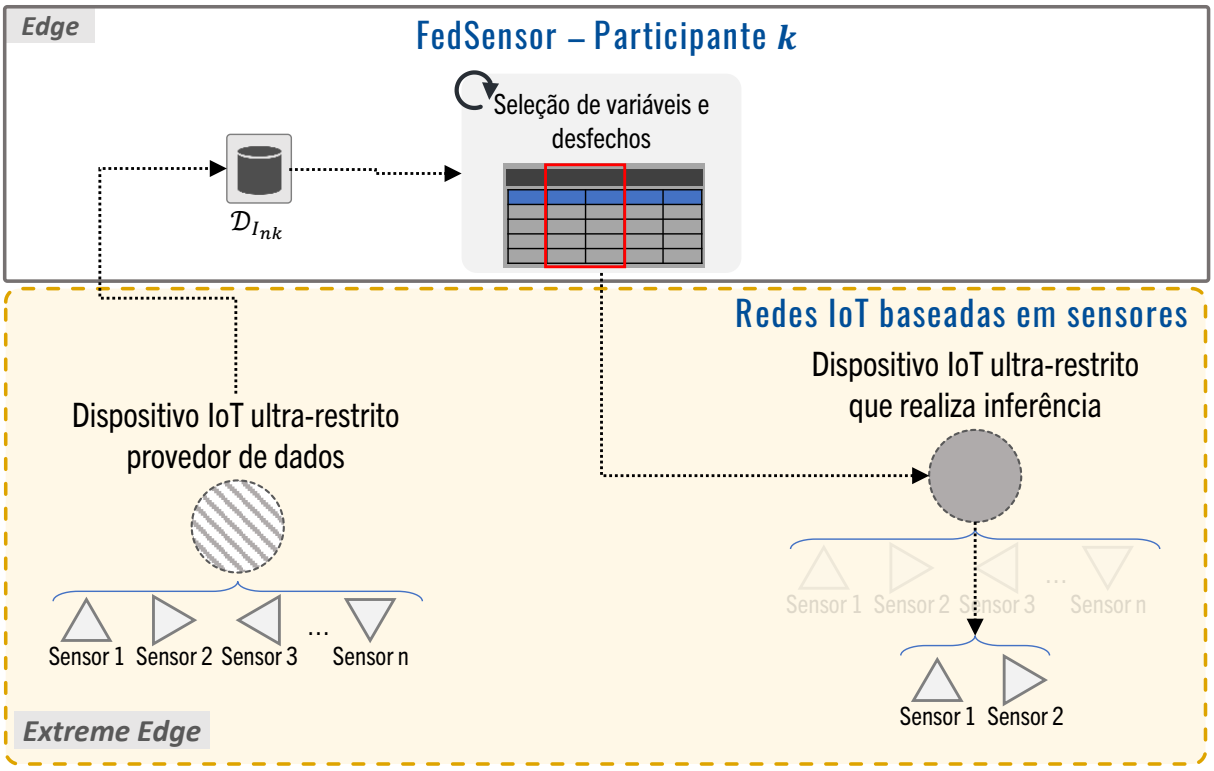
No FedSensor, escolher um novo conjunto de variáveis implica em coletar (ou deixar de coletar) novas medições dos sensores dos dispositivos para a realização da inferência. É importante ressaltar que há dois tipos de dispositivos IoT ultra-restritos provisionados no FedSensor: (1) os que enviam as medições, e (2) os que realizam a inferência. Quando o FedSensor define um novo modelo global, apenas os dispositivos IoT ultra-restritos que realizam a inferência para tomada de decisão têm o conjunto de variáveis alterado. Já os dispositivos publicadores de dados continuam a coletar e enviar as medições de todos os seus sensores. A razão desta prática é porque o envio de aproximadamente 100 medições diárias dos sensores dos dispositivos mantém a vida útil da bateria dos dispositivos próximo da vida útil com o dispositivo em descanso (FERRAZ JUNIOR et al., 2021a; FERRAZ JUNIOR et al., 2022).

A utilização de dispositivos IoT que realizam inferências e tomam decisões, não substitui o importante papel dos dispositivos que coletam as medições do ambiente em que estão inseridos e enviam essas medições para o *Edge Server*. Pois para que os modelos de ML continuem sendo aperfeiçoados são necessárias novas medições constantemente coletadas. Essas novas medições viabilizam o treinamento do modelo local e, por consequência, o modelo global. Com o conjunto de dados no *Edge Server* sendo alimentado continuamente, pode-se realizar nova análise de seleção de variáveis no futuro - e, dependendo dos dados, pode ser formado um outro conjunto de variáveis selecionadas. Esse fluxo é apresentado na Figura 10.

Na Figura 10, observa-se o dispositivo IoT ultra-restrito provedor de dados utilizando n sensores, enquanto o dispositivo IoT ultra-restrito que realiza a inferência usando apenas os sensores definidos pelo módulo *Seleção de variáveis e desfechos* do FedSensor.

Nesse sentido, observa-se a importância da cooperação nuvem-*Edge*-*Extreme edge*, para que os dispositivos IoT ultra restritos possam receber novos modelos de ML e realizar a inferência. Sem o uso da camada *Edge*, os dispositivos ultra-restritos ficariam impossibilitados de se integrar em arquiteturas tradicionais de FL, pois os modelos não são desenhados para dispositivos com severas restrições, bem como seria inviável a avaliação de novos modelos para as aplicações IoT existentes no FedSensor, pois há sucessivas iterações

Figura 10 – Seleção de variáveis aplicada ao dispositivo IoT ultra-restrito.



Fonte: Autor

entre nuvem-*Edge* para a definição do melhor modelo, bem como para o treinamento.

Como se utilizam dispositivos IoT ultra restritos em redes IoT baseadas em sensores (cujos dispositivos podem estar em locais remotos sem assistência humana), a energia é um recurso escasso e a bateria deve durar o maior tempo possível sem que precise ser substituída. A seleção de variáveis, portanto, viabiliza a redução do consumo de energia.

Após a definição do novo modelo de ML global, o gerenciador envia esse novo modelo aos participantes. Ao receber o novo modelo de ML global, o *Edge Server* passa a enviar as mensagens de configuração do novo modelo para os dispositivos IoT ultra-restritos conforme apresentado na próxima subseção.

4.6 Transmissão do modelo global para os dispositivos IoT ultra-restritos

Um dos objetivos elencados neste Tese se refere à transmissão do modelo de ML para os dispositivos IoT. Como resultado é desenvolvido o LWPubSub, conforme se observa nas seguintes publicações: Ferraz Junior et al. (2021a), Ferraz Junior et al. (2021b), e Ferraz

Junior et al. (2022). O LWPubSub é um sistema de mensagens sensíveis ao contexto com segurança fim-a-fim para a transmissão de dados entre dispositivos IoT ultra-restritos e um *Edge Server*, usando mensagens "publish-subscribe".

O LWPubSub se mostra eficiente energeticamente, pois requer o menor número de bytes no tópico e carga útil de mensagens *publish-subscribe*, culminando em um baixo consumo de energia para os dispositivos IoT ultra-restritos (FERRAZ JUNIOR et al., 2021b; FERRAZ JUNIOR et al., 2022). Portanto, neste trabalho o LWPubSub é utilizado de duas maneiras: (1) para os dispositivos publicadores de dados (que enviam suas medições para o participante), e (2) para os dispositivos que realizam a inferência.

A padronização utilizada no LWPubSub permite que os múltiplos objetos (sensores de temperatura, umidade, acelerômetro; atuadores, como leds; entre outros) dos dispositivos tenham um identificador único, denominado *objectID*. Cada objeto pode ter uma ou mais instâncias, denominadas *instanceID*. Dessa forma, é possível obter as informações dos dispositivos de maneira padronizada e identificando unicamente os sensores (mesmo que do mesmo tipo), usando os metadados no formato "objectID/instanceID".

Por exemplo, dois sensores de temperatura de um dispositivo utilizam o mesmo objectID, mas cada sensor corresponde a um instanceID diferente. Seguindo o padrão IPSO, esses dois sensores de temperatura são referenciados da seguinte maneira: 3303/0 e 3303/1, onde 3303 é o objectID referente ao sensor de temperatura e 0 e 1 referenciam cada sensor do dispositivo. Outros objectID comumente utilizados são umidade (3304) e led (3311). Uma mensagem que acende um led pode ser: 3311/0 1, onde 1 se refere à implementação no sensor de que o led deve ser ligado.

O LWPubSub pode, portanto, ser utilizado para a transmissão padronizada de mensagens entre o dispositivo IoT ultra-restrito e o *Edge Sever*, tanto para os dispositivos transmitirem os dados coletados pelos sensores, quanto para o *Edge Sever* transmitir o modelo de ML global. Para isso, o *objectID* identifica o modelo a ser utilizado e o *instanceID* identifica a quantidade de coeficientes e seus valores.

A transmissão dos modelos de ML globais usando o LWPubSub contempla o envio de duas mensagens.

A *mensagem 1* indica quais sensores do dispositivo serão habilitados e qual a frequência de obtenção das medições dos sensores. Isso porque os dispositivos IoT podem

ser utilizados em diferentes aplicações IoT, as quais podem requerer várias medições por minuto (caso dos *wearables*) ou poucas medições por dia (caso de medições do ambiente). A *mensagem 2* configura, no dispositivo, qual modelo de ML deve ser utilizado e os parâmetros/coeficientes (os modelos suportados devem estar no firmware do dispositivo IoT). Para modelos de agrupamento ou classificação, além de enviar os coeficientes do modelo, a mensagem 2 indica o grupo ou a classe que dispara a ação a ser tomada. No caso de modelos de regressão, a mensagem 2 carrega o valor corresponde ao valor máximo aceitável, que se ultrapassado dispara a ação a ser realizada pelo dispositivo.

Os modelos de ML têm características diferentes, mas que permitem a transmissão dos seus parâmetros para o dispositivo IoT ultra restrito. A seguir apresentam-se as formas de geração das mensagens e transmissão dos modelos na *mensagem 2*.

4.6.1 Modelos de regressão no FedSensor

No caso de uma *regressão linear simples* (JAMES et al., 2013) (RLS), há a modelagem de previsão de uma variável resposta Y quantitativa baseando-se em uma única variável preditora X. A partir dos dados de treinamento, estimam-se os coeficientes β_0 e β_1. Apresenta-se a RLS na Eq. 4.8.

$$Y = \beta_0 + \beta_1 X \tag{4.8}$$

A *mensagem 2* em uma RLS tem a seguinte composição:

- *objectID*: `32101`.
- *instanceID* `0`: valor máximo aceitável.
- *instanceID* `1`: β_0.
- *instanceID* `2`: β_1.

Uma *regressão linear múltipla* (JAMES et al., 2013) (RLM) expande a RLS permitindo mais variáveis preditoras, sendo representada conforme a Eq. 4.9.

$$Y = \beta_0 + \beta_1 X_1 + ... + \beta_p X_p \tag{4.9}$$

No caso da RLM, a mensagem de definição/atualização de modelo de ML segue o padrão da RLS, mas tem como código o *objectID* `32102`.

4.6.2 Modelos de classificação no FedSensor

Em modelos com respostas qualitativas, normalmente emprega-se o classificador de Bayes (JAMES et al., 2013), aplicando o Teorema da Probabilidade Condicional. O classificador de Bayes atribui cada observação à classe mais próxima, dado os valores das variáveis preditoras - uma observação com um vetor de valores preditores x_0 é atribuída a uma classe j conforme a Eq. 4.10.

$$Pr(Y = j|X = x_0) \tag{4.10}$$

Por exemplo, em um problema com apenas duas classes (classe 1 e classe 2) pode-se definir o classificador de Bayes para resultar na previsão para classe 1 caso $Pr(Y = 1|X = x_0) > 0,5$ e para a classe 2 no caso contrário. Para isso, deve-se estimar a distribuição condicional de Y dado X.

O modelo *regressão logística* (JAMES et al., 2013) é um exemplo de classificador que pode se basear no classificador de Bayes.

No caso da variável dependente Y assumir dois estados (0 ou 1) em um conjunto de p variáveis independentes $(X_1, X_2, ..., X_p)$, o modelo de regressão logística tem a forma apresentada na Eq. 4.11.

$$P(Y = 1) = \frac{e^{\beta_0+\beta_1 X_1+...+\beta_p X_p}}{1 + e^{\beta_0+\beta_1 X_1+...+\beta_p X_p}} \tag{4.11}$$

Usando o FedSensor, o modelo e os parâmetros β_0, β_1, ..., β_p são transmitidos da mesma forma que em uma RLM, mas usando o *objectID* `32103` na mensagem de configuração do modelo.

4.6.3 Modelos de agrupamento no FedSensor

A intenção do aprendizado não-supervisionado muitas vezes está relacionada à identificação de grupos similares das observações coletadas.

Uma das técnicas de aprendizado não-supervisionado e que pode ser utilizada no FedSensor são os agrupamentos (clustering) usando o K-means.

O agrupamento por K-means resulta no particionamento das observações em k grupos distintos não-sobrepostos. Para isso, é necessário definir o número de grupos (*clusters*). Está fora do escopo desta Tese a definição do melhor K.

O K-means utiliza medidas de dissimilaridade intra-cluster e inter-cluster. Dentre as medidas de dissimilaridade, a mais aplicada para o K-means, segundo James et al. (2013) é a distância Euclidiana (Eq. 4.12).

$$d(x, y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \tag{4.12}$$

Cada grupo tem um elemento central, denominado centróide, que é atualizado segundo a Equação 4.13. A atualização dos centróides ocorre até que o centróide não mude de posição em duas iterações sucessivas.

$$C_i = \frac{1}{||S_i||} \sum_{x_j \in S_i} x_j \tag{4.13}$$

Na Equação 4.13, C_i é o i-ésimo centróide, S_i são todos os pontos pertencentes ao i-ésimo conjunto com centróide C_i, x_j é o j-ésimo ponto do conjunto e $||S_i||$ é o número de pontos no i-ésimo conjunto.

Para o FedSensor, os parâmetros transmitidos são os valores dos centróides, usando o *objectID* 32105.

4.6.4 Características gerais da transmissão

Os modelos de ML transmitidos usando o FedSensor requerem a serialização dos parâmetros para envio aos dispositivos IoT ultra-restritos. Consideram-se que os parâmetros do modelo (as variáveis preditoras) enviados na mensagem 2, correspondem aos sensores previamente ativados nos dispositivos na mensagem 1.

Nesse sentido, uma atualização de modelo de ML global que mantém os mesmos sensores em operação não requer o envio da mensagem 1 - mesmo que outro modelo de ML seja utilizado, por exemplo, com uma troca entre regressão logística e k-means. Dessa forma, a utilização de outros sensores requer o envio das mensagens 1 e 2.

Independentemente do modelo de ML global utilizado na aplicação IoT, conforme se observa no Capítulo 2, a eficiência energética é uma questão a ser observada em arquiteturas

que distribuem para borda das redes IoT a tarefa de processamento (que antes ocorria unicamente na nuvem) - caso das camadas *edge* e *extreme edge*. O consumo de energia é abordado na seção a seguir.

4.7 Consumo de energia

O FedSensor atribui para as camadas *edge* e *extreme edge* as principais tarefas relacionadas à geração de novos modelos de ML em uma arquitetura de FL.

Por isso é necessário avaliar o consumo de energia para as ações de treinamento (que corre nos participantes) e de inferência (que ocorre nos dispositivos IoT ultra-restritos). O consumo de energia é crucial, pois um dispositivo IoT ultra-restrito que utiliza o FedSensor, deve gastar energia de acordo com as próprias características, sem que isso inviabilize o uso do dispositivo.

No FedSensor as rodadas de treinamento são executadas constantemente, com a intenção de fornecer mais acurácia para o modelo de ML global, enviados a cada rodada para os dispositivos IoT ultra-restritos. Dessa maneira, o consumo de energia dessas constantes trocas de modelo de ML global devem ser analisadas para avaliar o comportamento dos dispositivos frente às constantes atualizações dos modelos e realização de inferência.

O consumo de energia, portanto, é avaliado de duas maneiras: (1) na *edge*, e (2) na *extreme edge*.

Na *edge* a análise do consumo de energia baseia-se na capacidade e no tempo que o participante requer para realizar as seguintes ações (i): o treinamento do modelo de ML local, a transmissão do modelo de ML local para a nuvem, o recebimento do modelo de ML global, a transmissão do modelo de ML global para os dispositivos IoT ultra-restritos. Deve-se considerar, também, o tempo que o participante gasta estando em inatividade. A Equação 4.14 apresenta o cálculo para um participante k.

$$E_{edge} = \sum_{i=1}^{n} P_i \times t \tag{4.14}$$

Onde P é o consumo em mW, t é o tempo gasto nas i ações anteriormente descritas.

O consumo de energia nos dispositivos IoT ultra-restritos, por outro lado, precisa ser

mais granular, considerando as restrições dos dispositivos, conforme se observa em Ferraz Junior et al. (2022), Jalali et al. (2016). Nesse sentido, a equação 4.15 apresenta o modo de avaliação do consumo de energia nesses dispositivos.

$$E_{dispositivos} = \sum_{a=1}^{A} \sum_{i=1}^{n} t_i \times V \times C_i \tag{4.15}$$

Onde V é a voltagem, t é o tempo gasto nos i recursos do dispositivo $i = (CPU, LPM, TX, RX)$, C é a corrente elétrica e a corresponde às seguintes ações executadas pelo dispositivo: inativo, inferência, recebimento dos sensores a serem ativados (mensagem 1), e recebimento do modelo global (mensagem 2).

4.8 Síntese

Nos dispositivos IoT ultra restritos, o FedSensor tem a função de realizar a inferência (fase de testes) do modelo de ML. Isso implica que a tomada de decisão ocorre no próprio dispositivo e não aguarda uma decisão advinda da plataforma em nuvem. Considera-se como escopo que o modelo de ML (e seus parâmetros) utilizado para a tomada de decisão pode ser alterado durante a utilização do dispositivo.

O LWPubSub é um sistema de mensagens seguras fim-a-fim que serializa o modelo de ML global para que o dispositivo possa recebê-lo e passar a coletar as medições dos sensores para realizar a inferência. Os sensores dos dispositivos IoT ultra-restritos são as variáveis dos modelos de ML.

A proposta desta Tese difere da apresentada nos trabalhos de Warden e Situnayake (2020), Sliwa, Piatkowski e Wietfeld (2020), Lin et al. (2020), Kumar, Goyal e Varma (2017) e Wang, Li e He (2019). Os citados trabalhos tem o propósito de implantar nos dispositivos ultra restritos um modelo de ML fixo para tomada de decisão. Além disso, esta Tese não tem a intenção de implantar modelos de ML nos dispositivos da camada *extreme edge* com o intuito de avaliar acurácia, precisão ou outros fatores intrínsecos aos modelos de ML. A proposta é permitir que o modelo de ML usado no dispositivo seja modificado com o dispositivo em execução, com a atualização dos parâmetros (coeficientes ou preditores) de ML e seus respectivos valores. Além disso, a proposta desta Tese não limita (nem lista) os modelos existentes e que podem ser suportados pelo FedSensor.

Dessa forma, os resultados esperados de privacidade e anonimidade dos dispositivos, assim como a eficiência energética são possíveis de serem atingidos.

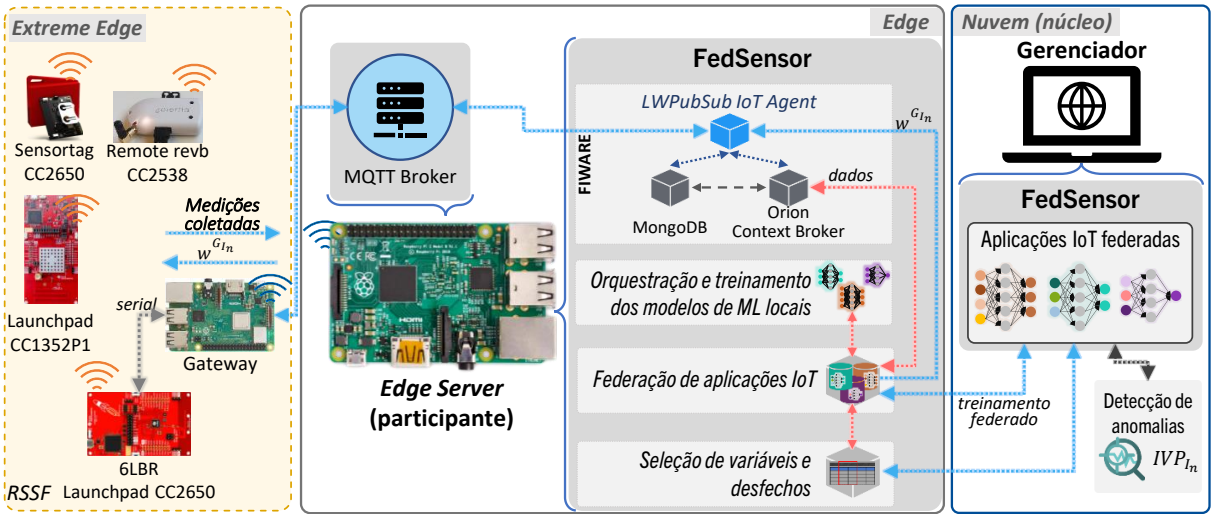
5 EXPERIMENTOS

Neste capítulo apresentam-se os experimentos conduzidos em uma arquitetura de FL que contempla as camadas nuvem, *Edge*, *Extreme Edge* e os dispositivos IoT ultra-restritos. Também apresentam-se os conjuntos de dados avaliados, os modelos de ML e as características do sistema de mensagens.

5.1 Características gerais dos experimentos

Nos experimentos deste trabalho, utiliza-se um sistema de FL, considerando duas vertentes: (1) uma avaliação do sistema de mensagens e tráfego de dados entre os dispositivos IoT ultra-restritos e o servidor na camada Edge, e (2) uma avaliação do treinamento dos modelos de ML aplicando a proposta apresentada no FedSensor (federação de aplicações IoT, detecção de anomalias e seleção de variáveis).

Figura 11 – Visão geral dos experimentos realizados



Fonte: Autor

O cenário experimental relativo ao hardware e a integração entre os diferentes componentes do FedSensor é o apresentado na Figura 11. Nesse cenário, há um servidor (Gerenciador), representando a camada da nuvem, e que realiza a agregação dos modelos globais. A principal camada de inteligência do FedSensor está no *Edge Server* (Participante), que compreende uma *microcloud* executada em um Raspberry Pi modelo 3 B, e, por

fim, dispositivos IoT ultra-restritos implantados em uma rede IoT baseada em sensores (6LoWPAN), representando a camada *Extreme Edge*.

Apresenta-se, na Tabela 2, os componentes e recursos de hardware, software, sensores e conectividade.

Tabela 2 – Componentes e recursos das camadas nuvem, *Edge* e *Extreme edge* do FedSensor

Gerenciador			
Hardware	Core i7, 16GB RAM, GTX 1650, 1TB HDD		
Participante			
Hardware	Rasperry Pi Modelo 3 B, 1.2 GHz CPU, 1 GB RAM		
Serviços IoT na *Edge*	Baseada na plataforma FIWARE		
MQTT Broker	*mosquitto* MQTT Broker		
6LBR	Firefly CC2538 acoplado a um Raspberry Pi Modelo 3 B		
Context-Broker	Orion 2.3.0		
Banco de dados	Mongo DB 3.6		
LWPubSub IoT Agent	Desenvolvido pelo autor a partir do IoT Agent da FIWARE		
Mensagem LWPubSub	Desenvolvida pelo autor		
Treinamento federado	Desenvolvido pelo autor com base no *flower* (BEUTEL et al., 2020)		
Dispositivos IoT ultra-restritos			
Recurso	**Sensortag**	**Remote**	**CC1352P1**
TX	6.10 mA	24 mA	7.10 mA
RX	5.90 mA	20 mA	6.90 mA
Low Power Mode (LPM)	0.55 mA	0.60 mA	0.59 mA
CPU	2.97 mA	13 mA	2.89 mA
Microcontrolador	CC2650 Cortex M3	CC2538 Cortex M3	CC1352 Cortex M4
ROM	128 KB	512 KB	352 KB
RAM	20 KB	32 KB	80 KB
Sistema operacional	Contiki-NG v.4.7 (OIKONOMOU et al., 2022)		

No caso da camada Extreme Edge, conforme apresentado na Tabela 2, os dispositivos IoT ultra-restritos CC1352P1, Remote e Sensortag, se utilizam do sistema operacional Contiki-NG, um sistema operacional de código aberto, multi plataforma e desenvolvido para dispositivos embarcados com severas restrições de processamento, memória, armazenamento e energia (ultra-restritos). O propósito do Contiki-NG é viabilizar para os dispositivos IoT ultra-restritos o uso dos protocolos de aplicação padrão na IoT, como MQTT e CoAP, baseando-se nas comunicações em redes 6LoWPAN.

O Contiki-NG também contém módulos para apoiar a observação do consumo de energia dos dispositivos. O cálculo do consumo de energia desprendido pelos dispositivos IoT ultra-restritos segue o disposto na Equação 4.15, apresentada no Capítulo 4. O

tempo desprendido em cada tarefa é obtido por meio do módulo Energest do Contiki-NG (DUNKELS et al., 2007).

Para a camada *Edge*, no participante, o treinamento do modelo de ML local e a interação com o gerenciador na nuvem para geração do modelo de ML global ocorre por meio do treinamento federado, sendo desenvolvido um módulo para integração das diferentes aplicações IoT apresentadas. O treinamento federado é realizado por aplicação IoT e, quando concluído, todos os participantes recebem o modelo de ML global e o transmitem para os dispositivos IoT ultra-restritos referentes àquela aplicação IoT.

Ainda no participante, com relação à transmissão das mensagens, máquinas virtuais baseadas em contêineres executam um MQTT Broker. Utiliza-se o sistema de mensagens LWPubSub, desenvolvido e publicado durante o desenvolvimento desta Tese em Ferraz Junior et al. (2021a), Ferraz Junior et al. (2021b), Ferraz Junior et al. (2022) para a transmissão de dados entre os dispositivos e o *Edge Server*. Neste trabalho, o LWPubSub transporta tanto os dados coletados dos sensores, quanto os os modelos de ML globais, considerando as características de cada dispositivo. Para a execução de experimentos, considera-se no escopo deste trabalho o uso de padrões abertos. Nesse sentido, a plataforma fornecida pela FIWARE [1] fornece a infraestrutura necessária para uma plataforma em nuvem aberta usada nas camadas *edge* e *nuvem*. A plataforma FIWARE baseia-se em recursos denominados Generic Enablers (GE), os quais fornecem os diversos serviços que compõem uma plataforma em nuvem. Um dos principais GE para IoT é o IoT Agent: um serviço existente para transmitir e receber as mensagens dos dispositivos IoT. Além disso, outro importante GE é o que fornece serviço de contexto às mensagens: o Orion Context-Broker.

Com relação à transmissão de mensagens, assim como explorado nos Capítulos 2 e 3, existem diferentes protocolos de comunicação *publish-subscribe* que podem ser aplicados na comunicação entre os dispositivos IoT e o *Edge Server*. Na sequência apresenta-se a justificativa do uso do MQTT como protocolo *publish-subscribe* a ser utilizado no FedSensor.

[1] https://www.fiware.org/about-us/

5.1.1 Avaliação do LWPubSub

O trabalho de Feraudo et al. (2020) se utiliza do MQTT para o tráfego de modelos de ML entre um servidor e dispositivos IoT robustos, como SBC. De maneira a subsidiar a decisão do sistema de mensagens a ser utilizado para transmissão de mensagens entre um *Edge Server* e um dispositivo IoT ultra-restrito, neste trabalho avaliam-se três protocolos *publish-subscribe*: MQTT, AMQP e DDS.

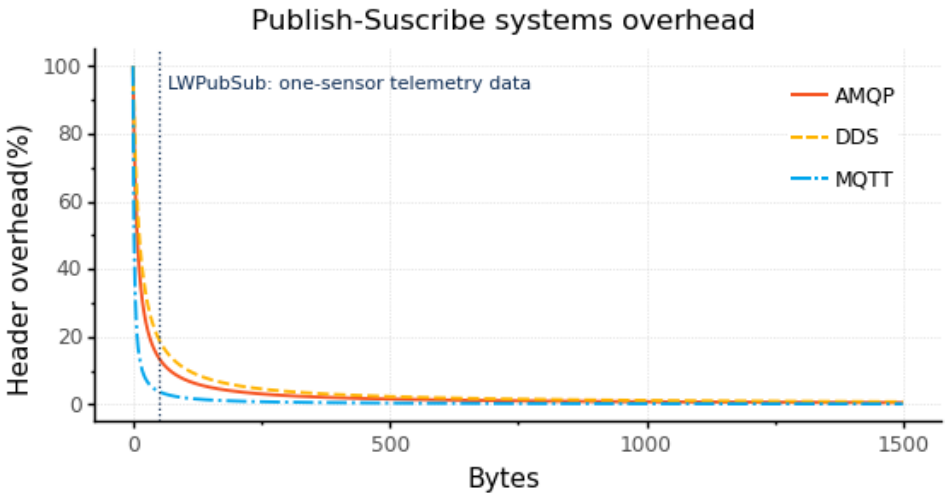
Com relação à transmissão de medições coletadas pelos dispositivos e de comandos recebidos para a execução de ações pelo dispositivo, a mensagem LWPubSub requer 16 bytes para enviar medições e 20 bytes para receber comandos do *Edge Server*. Considerando as características dos protocolos MQTT, AMQP e DDS, para enviar dados de telemetria de um sensor (como temperatura, umidade e acelerômetro, entre outros), a carga útil com segurança fim-a-fim tem 32 bytes de tamanho. Portanto, a mensagem LWPubSub (que engloba tópico+carga útil) requer 52 bytes.

Há diferenças entre a sobrecarga de cabeçalho dos protocolos AMQP, DDS e MQTT quando se usa o LWPubSub. Apresenta-se a seguir o percentual de sobrecarga de cabeçalho em relação ao tamanho da mensagem transmitida. Calcula-se o percentual de sobrecarga de cabeçalho conforme a Equação 5.1, considerando como tamanho de bytes do cabeçalho: 2, 8 e 12 bytes, respectivamente para MQTT, AMQP e DDS. Resultados preliminares publicados durante esta Tese - em Ferraz Junior et al. (2022) - demonstram que o MQTT é o protocolo que oferece menor sobrecarga na relação tópico-carga útil, conforme se observa na Figura 12.

$$\text{Sobrecarga cabeçalho}(\%) = \frac{\text{Cabeçalho}}{\text{Cabeçalho+Carga útil}} \times 100 \tag{5.1}$$

Com relação à segurança na transmissão de mensagens, dispositivos IoT robustos, como os SBC, podem usar uma suíte de protocolos que se utilize dos mesmos algoritmos usados em desktops e servidores convencionais, como o TLS, por exemplo. Entretanto, dispositivos IoT ultra-restritos não são capazes de executar o TLS (FERRAZ JUNIOR et al., 2021a). Ainda assim, mesmo usando TLS, a carga útil precisa de segurança adicional para não revelar os dados para qualquer outra entidade que não seja o destinatário da mensagem - caso a carga útil não tenha segurança, o Broker pode ler as mensagens. Consequentemente, um mecanismo de segurança deve existir para proteger os dados,

Figura 12 – Relação cabeçalho-carga útil entre os protocolos MQTT, AMQP e DDS



Fonte: Ferraz Junior et al. (2022)

independentemente do sistema de mensagens publish-subscribe utilizado, uma vez que o Broker é um intermediário nessa comunicação.

Diante do apresentado, a utilização do LWPubSub apresenta uma nova forma de composição do tópico e da carga útil para transmissão de dados entre dispositivo IoT ultra-restrito e a Edge, fornecendo, além da segurança fim-a-fim, padronização e eficiência energética Ferraz Junior et al. (2021a), Ferraz Junior et al. (2021b) e Ferraz Junior et al. (2022). Constata-se eficiência energética tanto para os dispositivos (a mensagem LWPubSub aumenta a vida útil das baterias) quanto para as plataformas *Edge* (pois reduz o número de bytes trafegados).

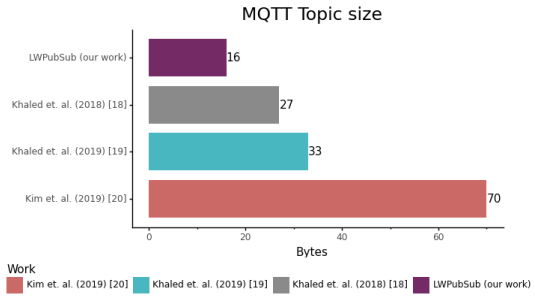
O LWPubSub estrutura as mensagens usando uma identificação única para cada dispositivo provisionado no participante(*deviceID*). O tópico */domínio/deviceID* unicamente identifica um dispositivo em um *Edge Sever*.

Com relação aos sensores de um dispositivo IoT ultra-restrito, usa-se o registro fornecido pela IPSO (OMA LightweightM2M (LWM2M), 2017), aplicando a tupla *objectID/instanceID* para unicamente identificar os sensores dos dispositivos - o *objectID* é o tipo do sensor, enquanto o *instanceID* o número do sensor daquele tipo utilizado no dispositivo (pois é possível ter mais de um sensor do mesmo tipo - como por exemplo dois sensores de temperatura, assim instanceID identifica o sensor utilizado).

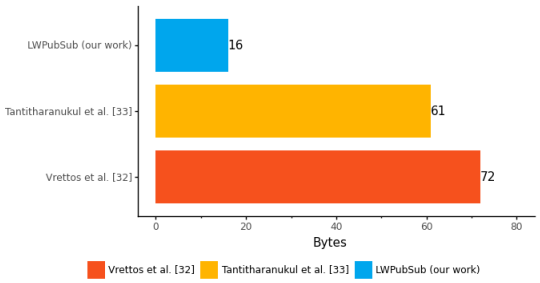
O uso do LWPubSub se mostra o cenário mais adequado para a entrega de mensagens seguras fim-a-fim entre um *Edge Server* e um dispositivo IoT ultra-restrito, uma vez que o tópico e a carga útil apresentam o menor número de Bytes necessários para o tráfego de mensagens.

Com relação ao tópico das mensagens, a Figura 15 apresenta os resultados alcançados com o uso do LWPubSub, que requerem menos Bytes quando comparados aos principais trabalhos relacionados.

Figura 13 – Consumo de Bytes do tópico das mensagens LWPubSub



(a) Comparação do tamanho do tópico de mensagens LWPubSub com trabalhos relativos à sensibilidade ao contexto



(b) Comparação do tamanho do tópico de mensagens LWPubSub com trabalhos relativos ao tráfego de dados de aplicações IoT

Fonte: Ferraz Junior et al. (2021a) e Ferraz Junior et al. (2022), respectivamente

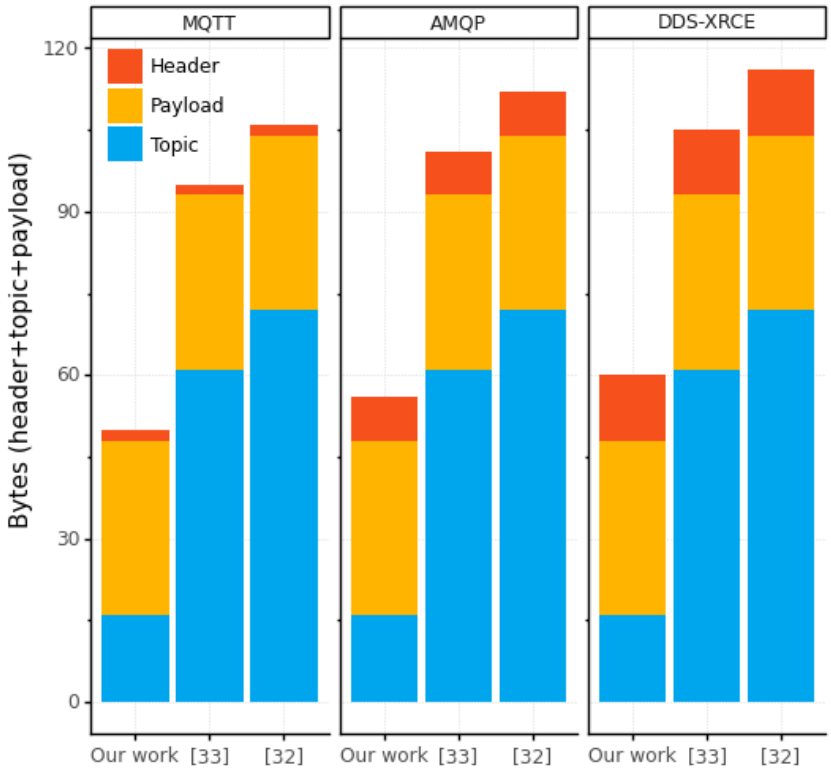
Além disso, a Figura 14 comprova que o MQTT apresenta o menor consumo de Bytes para a transmissão de mensagens seguras fim-a-fim entre um dispositivo IoT ultra-restrito e um *Edge Server*.

Considerando o exposto, observa-se que o LWPubSub é um sistema de mensagens que avança no estado-da-arte para a transmissão de mensagens sensíveis ao contexto e seguras fim-a-fim. Por isso, usa-se o LWPubSub para a transmissão dos modelos de ML globais do *Edge Server* para os dispositivos IoT ultra-restritos.

5.2 Cenários experimentais

Considerando a utilização do LWPubSub para a transmissão dos modelos de ML globais gerados no FedSensor, apresentam-se na Tabela 3 as características gerais aplicadas nos cenários experimentais: recursos do sistema de mensagens como domínio, dispositivos (e seus recursos de hardware e software), frequência de treinamento do modelo de ML

Figura 14 – Tamanho das mensagens MQTT, AMQP e DDS com a utilização do LWPubSub para a transmissão mensagens sensíveis ao contexto e seguras fim-a-fim



Fonte: Ferraz Junior et al. (2022)

global, frequência da realização de inferência pelos dispositivos e modelos de ML utilizados na arquitetura de FL proposta.

As aplicações IoT podem requerer diferentes intervalos de tempo na coleta das medições dos dispositivos, as quais podem variar desde o uso como *wearable* (que exige muitas medições por minuto) até o uso na medição de condições ambientais, as quais requerem poucas medições por dia. Portanto, para observar o comportamento dos dispositivos em diferentes condições de operação, os experimentos a serem conduzidos contemplam as frequências para obtenção de medições apresentadas na Tabela 3.

O intervalo de tempo para coleta das medições (frequência de realização de inferência pelos dispositivos IoT ultra-restritos, conforme consta na Tabela 3) considera experimentos anteriormente avaliados e publicados durante o desenvolvimento deste trabalho. Conforme publicado em Ferraz Junior et al. (2021a) e Ferraz Junior et al. (2022), o intervalo de tempo maior que 900 segundos não influencia no tempo de vida útil da bateria do dispositivo.

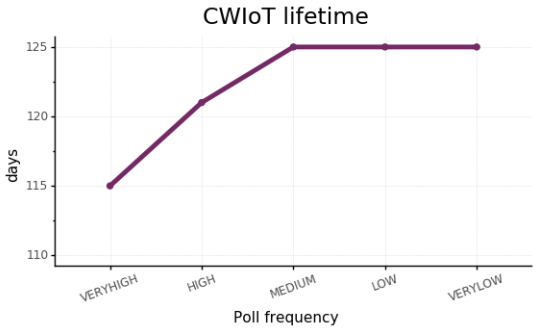
Tabela 3 – Parâmetros gerais do sistema de mensagens LWPubSub no FedSensor

Sinalização	
Domínio	99
Site (prefixo IPv6)	fd00::/64
CC1352P1 *deviceID*	00124ba1ad06
Remote revb *deviceID*	00124b4a527d
Sensortag *deviceID*	00124b05257a
objectID dos sensores	3313 (acelerômetro), 3325 (gases) e 3338 (alarme)
Tópico para receber comandos	/99/<*deviceID*>/cmd
Tópico para envio de medições	/99/<*deviceID*>
Algoritmo de segurança	AES-128-CCM-8
Frequência de atualização do modelo de ML global	1/8, 1/4, 1/2, 1, 2, 4, 8, 24 horas
Frequência de realização de inferência pelos dispositivos IoT ultra-restritos	2, 10, 30, 60, 240 segundos

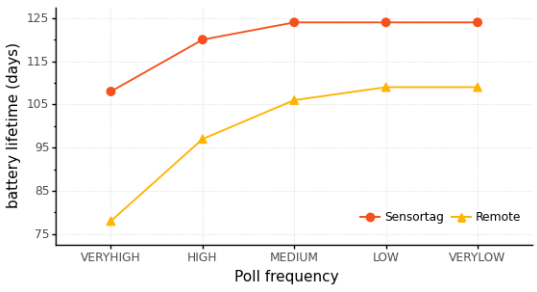
Modelos de ML avaliados		
Algoritmos	**ObjectID**	**Objetivo**
Regressão Linear	32102	valor máximo aceitável
Regressão Logística	32103	classe alvo
K-means	32105	centróide alvo

Nesse sentido, aumentar o intervalo de tempo para coleta de medições não traz benefícios, pois podem-se obter mais medições e realizar a tomada de decisão sem ter que aguardar um tempo demasiado, conforme apresentado na Figura 15. As Figuras 15a e 15b apresentam esses resultados do LWPubSub com relação ao tempo de vida útil da bateria dos dispositivos Sensortag e Remote. Pode-se notar que os intervalos de tempo “médio” (900 segundos), “alto” (21600 segundos) e “muito alto” (86400 segundos) apresentam o mesmo tempo de vida útil da bateria dos dispositivos.

Figura 15 – Duração das baterias dos dispositivos Sensortag e Remote com a utilização do LWPubSub para transmissão de mensagens seguras fim-a-fim com um *Edge Server*



(a) Consumo de energia do dispositivo Sensortag



(b) Consumo de energia dos dispositivos Sensortag e Remote

Fonte: Ferraz Junior et al. (2021a) e Ferraz Junior et al. (2022), respectivamente

Para avaliação das mensagens que carregam os modelos de ML globais para os dispositivos IoT ultra restritos, executam-se experimentos com 24 horas de duração para cada realização de inferência pelos dispositivos.

Diante do apresentado, para validação da proposta, aplica-se o FedSensor em duas aplicações IoT distintas (diferentes domínios): (1) Indústria 5.0, e (2) cidades inteligentes.

5.2.1 Validação do FedSensor na Indústria 5.0

Para avaliação do FedSensor na Indústria 5.0 define-se a aplicação de inteligência com relação à manutenção preditiva em um motor, assim como apresentado no trabalho de Sampaio et al. (2019).

Para o treinamento do modelo global no t_0, utilizam-se os dados disponibilizados por Sampaio et al. (2019). Os dados são segregados em três diferentes participantes, cada qual com uma parte do conjunto de dados $\mathcal{D}_{\mathcal{I}_{nk}}$. Posteriormente, os dispositivos IoT ultra-restritos apresentados na Tabela 2 recebem os novos modelos globais a cada vez que as iterações se encerram e passam a realizar a inferência.

A Tabela 4 apresenta os principais parâmetros do FedSensor e dos modelos globais avaliados no cenário Indústria 5.0.

Tabela 4 – Cenário experimental Indústria 5.0

Parâmetros da aplicação IoT: manutenção preditiva em motor	
Sensores	Acelerômetro 3 eixos (x, y, z)
objectID/instanceID dos sensores	33130 (eixo x), 33131 (eixo y), 33132 (eixo z)
Parâmetros dos modelos de ML avaliados	
Algoritmo	**Desfecho**
Regressão Logística	2 e 3 classes
K-means	2 e 3 grupos

A utilização de diferentes desfechos (grupos e classes divididos em dois e três), tem a intenção de estruturar e avaliar diferentes condições de operação dos modelos nos dispositivos. A utilização de 2 classes/grupos considera as possibilidades falha ou não-falha. A utilização de 3 classes/grupos considera as possibilidades do motor estar na posição normal (não-falha), oposta (falha) ou perpendicular (falha).

5.2.2 Validação do FedSensor em cidades inteligentes

Dentro do cenário de cidades inteligentes, o IQAr tem se mostrado foco de trabalhos recentes, em que o treinamento federado se coloca como um recurso fundamental ao permitir o aprendizado colaborativo mantendo a privacidade dos dados e, principalmente, dos dispositivos finais (ultra-restritos) que realizam a inferência.

Nesse sentido, os experimentos se utilizam de partições do conjunto de dados disponibilizado pela Índia[2] para construção do modelo de ML local em cada participante.

A Tabela 5 sumariza os parâmetros dos experimentos relativos ao IQAr.

Tabela 5 – Cenário experimental Cidades Inteligentes

Parâmetros da aplicação IoT: identificação do IQAr	
Sensores	Gases e partículas (PM 2.5, PM 10, NO, NO_2, NO_x, NH_3, CO, SO_2, O_3)
objectID/instanceID dos sensores	33250 (PM 2.5), 33251 (PM 10), 33252 (NO), 33253 (NO_2), 33254 (NO_x), 33255 (NH_3), 33256 (CO), 33257 (SO_2) e 33258 (O_3)
Parâmetros dos modelos de ML avaliados	
Algoritmo	**Desfecho**
Regressão Linear	valor máximo aceitável
Regressão Logística	2, 3 e 6 classes
K-means	2, 3 e 6 grupos

Os experimentos realizados com o IQAr permitem extrair as diferentes condições em que as aplicações IoT estão sujeitas. O conjunto de dados permite utilizar três diferentes modelos de ML: (1) regressão linear, (2) regressão logística, e (3) k-means), pois fornece tanto um valor numérico contínuo, quanto dividido em seis classes de acordo com a Tabela 6 relativa aos índices de IQAr, tendo como base os valores apresentados por (CHHIKARA et al., 2021).

Tabela 6 – Níveis de IQAr

Parâmetros da aplicação IoT: identificação do IQAr	
Nível	Valor
Perigoso (crítica)	301 - 500
Muito insalubre (péssima)	201 - 300
Insalubre (má)	151 - 200
Insalubre para grupos sensíveis (inadequada)	101 - 150
Moderada (regular)	51 - 100
Boa	0 - 50

[2] https://cpcb.nic.in/National-Air-Quality-Index/

Os modelos de classificação e agrupamento consideram modelos com duas classes/grupos: (1) qualidade do ar aceitável, ou (2) inaceitável. Três classes/grupos: (1) qualidade do ar boa, (2) aceitável, ou (3) inaceitável. Seis classes/grupos: (1) qualidade do ar boa, (2) regular, (3) inadequada, (4) má, (5) péssima, e (6) crítica, em que o alvo do modelo é identificar condições inaceitáveis ou críticas de IQAr. No caso da utilização da regressão linear, o limite mínimo que deflagra ação pelo dispositivo ao realizar a inferência é 300.

5.3 Seleção de variáveis

A seleção de variáveis no FedSensor tem como objetivo encontrar o conjunto de variáveis que propicie a máxima acurácia para o modelo de ML global de uma apliação IoT, considerando o apresentado no Capítulo 4.

Nesse sentido, considerando os experimentos apresentados, a seleção de variáveis aplica-se ao experimento *cidades inteligentes*, para identificar, dentre as nove variáveis possíveis do experimento *IQAr* quais são as que mais contribuem para o desfecho desejado do modelo de ML.

Para a seleção de variáveis, aplicam-se modelos de árvores de decisão para determinar as variáveis mais importantes para o desfecho. O uso de árvores de decisão, como florestas aleatórias, pode ser observado em (BELGIU; DRăGUţ, 2016; SUGUMARAN; MURALIDHARAN; RAMACHANDRAN, 2007).

O objetivo, em cada participante, é identificar as melhores variáveis de acordo com os dados existentes em cada $\mathcal{D}_{\mathcal{I}_{nk}}$.

A seleção de variáveis, com relação aos experimentos, tem duas vertentes:

- Nos dispositivos IoT ultra-restritos, observar o consumo de energia com a utilização de 2, 3, 4 e 9 sensores (que são as variáveis preditoras do modelo) e, adicionalmente para os modelos de ML de classificação e agrupamento, observar o consumo de energia com a utilização de 2, 3 e 6 classes/grupos;
- No participante, observar o consumo de energia considerando diferentes números de dispositivos IoT existentes em cada aplicação IoT gerenciada pelo participante. Avaliam-se aplicações IoT com 50, 100, 200 e 300 dispositivos IoT.

Além da seleção de variáveis, outro fator presente é a detecção de anomalias, apresentada no Capítulo 4 e que é detalhada na seção subsequente.

5.4 Detecção de anomalias

A detecção de anomalias é parte integrante do FedSensor, pois os dispositivos IoT que participam geram um grande volume de dados para as aplicações IoT existentes nos participantes.

Contudo, nos dados gerados podem ocorrer medições incorretas ou anômalas, oriundas de falhas nos sensores ou ataques. Essas medições anômalas induzem a erros na tomada de decisão resultante da aplicação de modelos de ML, como apresentam Rubin et al. (2020).

Por esse motivo, a detecção de anomalias atua para garantir a integridade dos dados. Rubin et al. (2020) apresentam, ainda, que o constante aumento da demanda na geração de dados na borda da rede faz com que a detecção de anomalias ocorra nos dispositivos da camada *Edge*, como é o caso dos participantes do FedSensor.

Considerando o apresentado no Capítulo anterior, a respeito da análise de variabilidade das funções de perda no gerenciador, é necessário considerar cenários experimentais que contenham participantes com e sem anomalias.

No caso dos participantes que contém anomalias, os experimentos consideram que esses participantes contêm medições anômalas em percentuais que variam de 30% a 70% dos valores normais, bem como alteração da classe dos resultados. Para observar o comportamento das anomalias frente à participantes que contenham apenas medições normais, utiliza-se o modelo de ML global de classificação (regressão logística).

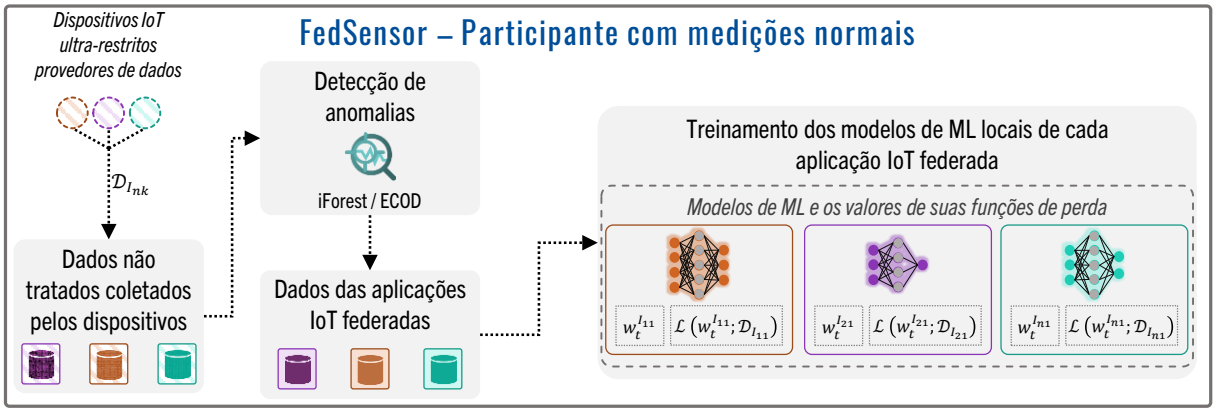
No caso dos participantes que contêm medições normais, o propósito é garantir que mesmo no conjunto de dados observado possíveis medições anômalas sejam retiradas. Por isso, duas técnicas para remoção de medições anômalas são utilizadas: (1) Isolation Forest (iForest), e Empirical Cumulative distribution functions for Outlier Detection (ECOD)

A análise apresentada por Rubin et al. (2020) conclui que a técnica iForest, dentre as técnicas avaliadas na camada *Edge*, é aquela que fornece maior acurácia na identificação de anomalias nos dados. Já o trabalho de Li et al. (2022) apresenta o ECOD, que tem por finalidade observar eventos raros nos dados, identificando anomalias. O trabalho de Li et

al. (2022) aponta o ECOD com uma acurácia média de identificação de anomalias maior quando comparado ao iForest (que fica em segundo na média dos conjuntos de dados avaliados).

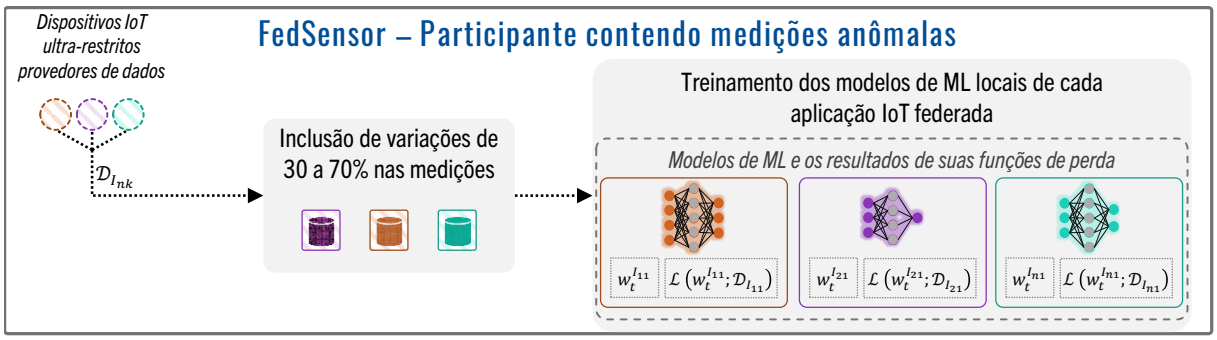
Diante do exposto, a Figura 16 mostra as tarefas realizadas por um participante que contém apenas medições normais, e a Figura 17 mostra as tarefas realizadas pelos participantes que incluem anomalias nas suas medições.

Figura 16 – Tarefas do experimento de um participante que contém medições normais, com a remoção de anomalias nas medições



Fonte: Autor

Figura 17 – Tarefas do experimento de um participante que contém medições anômalas, com a inclusão de anomalias nas medições



Fonte: Autor

Para a avaliação da detecção de anomalias, usam-se cinco participantes de treinamento federado com variações da presença de anomalias em 1, 3 ou em todos os participantes. As anomalias são geradas em percentuais que variam de 30 a 70% dos valores normais existentes nos conjuntos de dados das aplicações IoT avaliadas.

5.5 Síntese

Considerando os diferentes escopos de avaliação do FedSensor (consumo de energia nos dispositivos e no participante nos diferentes modelos de números de desfechos e de variáveis preditoras/sensores, detecção de anomalias, seleção de variáveis), realizaram-se os experimentos da seguinte maneira:

1. 360 transmissões da mensagem 1 do participante para os dispositivos IoT ultra-restritos CC1352P1, Remote e Sensortag;
2. 2250 transmissões da mensagem 2 do participante para os dispositivos IoT ultra-restritos CC1352P1, Remote e Sensortag;
3. 1250 rodadas de treinamento federado para geração dos modelos de ML globais avaliados;
4. 3150 rodadas de treinamento federado para a detecção de anomalias;
5. 1200 rodadas de treinamento federado para a avaliação do consumo de energia do participante (Raspberry Pi).

A quantidade de transmissões de dados e rodadas de treinamento federado apresentadas consideram cenários com a utilização de: (1) três modelos de ML globais (regressão linear, regressão logística e k-means), (2) três diferentes dispositivos IoT ultra-restritos (CC1352P1, Remote e Sensortag), (3) variação de intervalo de tempo para realização de inferência, (4) variação de intervalo de tempo para realização do treinamento federado e distribuição do modelo de ML global gerado para os participantes (e por conseguinte para os dispositivos), (5) variação do número de participantes com medições anômalas. Esse volume de experimentos subsidia analisar o consumo de energia e a identificação de participantes com medições anômalas no FedSensor.

A utilização de diferentes cenários experimentais considera a diversidade de aplicações IoT que precisam da realização da inferência em intervalos curtos ou longos, bem como a utilização de *testbed* composta por dispositivos IoT ultra-restritos em ambiente laboratorial permite obter as reais condições dos equipamentos em uma arquitetura de FL que une a camada nuvem, *Edge* e *Extreme Edge*. Com a realização dos experimentos, obtém-se os resultados apresentados e discutidos no capítulo a seguir.

6 RESULTADOS DO FEDSENSOR E DISCUSSÃO

A realização dos experimentos nos dois cenários propostos (Indústria 5.0, experimento *motor*, e cidades inteligentes, experimento *qualidade do ar*) permitem avaliar o FedSensor em reais condições de utilização que requerem: a privacidade e a anonimidade dos dispositivos IoT ultra-restritos, assim como a detecção de anomalias e a seleção das melhores variáveis para um desfecho. Os resultados dos experimentos são apresentados a seguir nos diferentes cenários apresentados no no Capítulo 5.

Os resultados visam observar o comportamento do FedSensor com relação às hipóteses levantadas e permitir extrair as conclusões referentes aos objetivos traçados, ambos apresentados no Capítulo 1.

No cenário *Indústria 5.0*, os modelos avaliados são o agrupamento com k-means e a classificação usando regressão logística, usando três sensores dos dispositivos (acelerômetro nos eixos x, y e z), pois a finalidade do desfecho é encontrar uma classe ou grupo que identifique um posicionamento de falha de um motor.

No cenário *Cidades inteligentes*, com o experimento *IQAr*, avaliam-se os modelos regressão linear, regressão logística e k-means, usando 2, 4 ou 9 variáveis (sensores) com três possibilidades de desfecho com relação à regressão logística e 2, 3 ou 6 grupos/classes para o k-means.

Considerando que milhares de dispositivos IoT ultra-restritos podem ser utilizados nas diferentes aplicações IoT, primeiro analisa-se o comportamento desses dispositivos com a utilização do FedSensor.

6.1 Consumo de energia para estruturação de sensores e variáveis dos modelos nos dispositivos IoT ultra-restritos

Conforme apresentado no Capítulo 4, para que os modelos de ML globais do FedSensor possam ser utilizados no FedSensor, os dispositivos IoT ultra-restritos precisam receber mensagens de para estruturação de quais sensores (que são as variáveis preditoras) devem ser utilizados pelo modelo de ML global. O consumo de energia dessa mensagem (a *mensagem 1* apresentada no Capítulo 4) corresponde ao gasto energético de CPU

(processamento), LPM (inativo), TX (transmissão) e RX (recepção) desprendido pelos dispositivos IoT ultra-restritos CC1352P1, Remote e Sensortag, calculados de acordo com a Equação 4.15 apresentada no Capítulo 4.

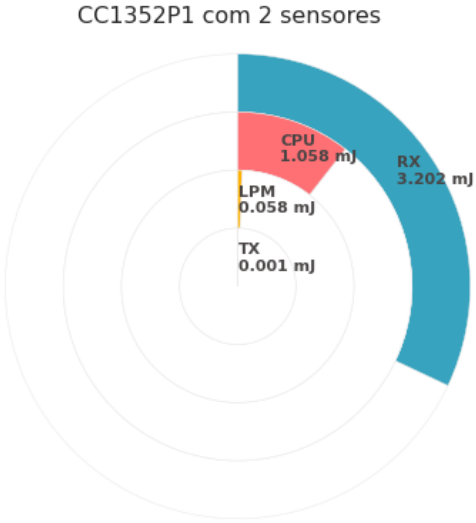
Ao se considerar as restrições de energia dos dispositivos IoT ultra-restritos, é importante observar qualquer mudança de comportamento, para assim poder extrair o consumo de energia de uma aplicação IoT, como é o caso do FedSensor. Por esse motivo, avalia-se o consumo de energia com o uso de diferentes números de variáveis preditoras dos modelos de ML globais.

As Figuras 18, 19 e 20 apresentam o consumo de energia em cada um dos dispositivos IoT ultra-restritos e o quanto de energia (em miliJoules - mJ) é requerido nos cenários IoT propostos, com a utilização de 2, 3, 4 ou 9 sensores nos dispositivos.

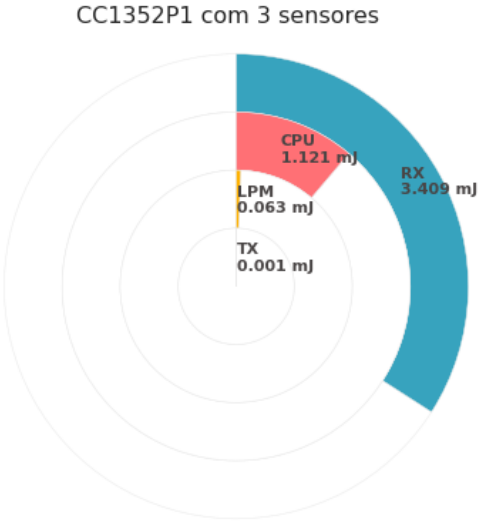
De acordo com o apresentado nas Figuras 18, 19 e 20, observa-se que o aumento do consumo de energia está relacionado à recepção (RX) das variáveis (sensores), enquanto o processamento (CPU) da mensagem apresenta percentual médio de 33% inferior (+- 0.07%).

Com relação ao número de sensores no próprio dispositivo, observa-se um aumento do consumo de energia de acordo com o aumento do número de sensores utilizados, embora essa mudança seja significativa apenas quando se usam 9 sensores, enquanto o uso de 2, 3 e 4 sensores apresentam consumo de energia similar.

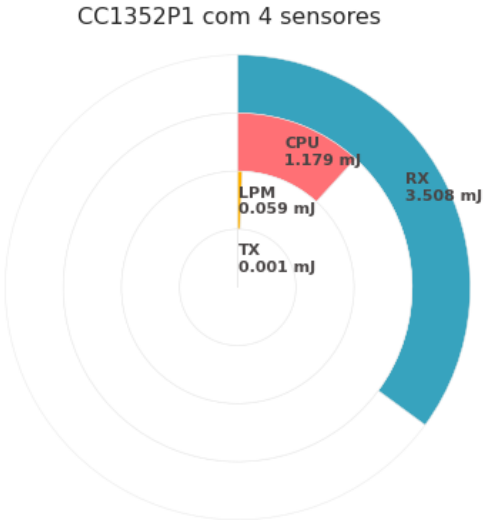
Figura 18 – Consumo de energia do dispositivo CC1352P1 para o recebimento da *mensagem 1* para estruturação das variáveis do modelo de ML global



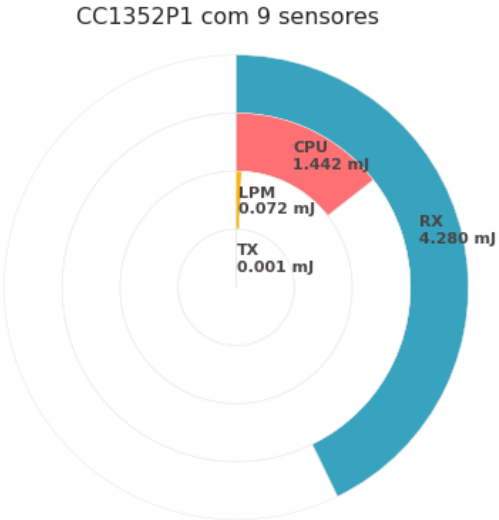
(a) Modelo de ML global usando 2 variáveis preditoras (sensores)



(b) Modelo de ML global usando 3 variáveis preditoras (sensores)



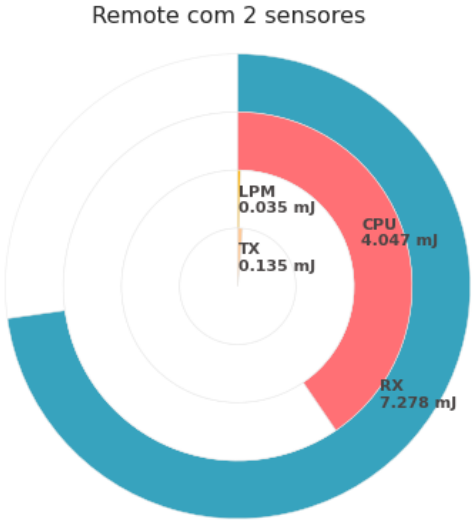
(c) Modelo de ML global usando 4 variáveis preditoras (sensores)



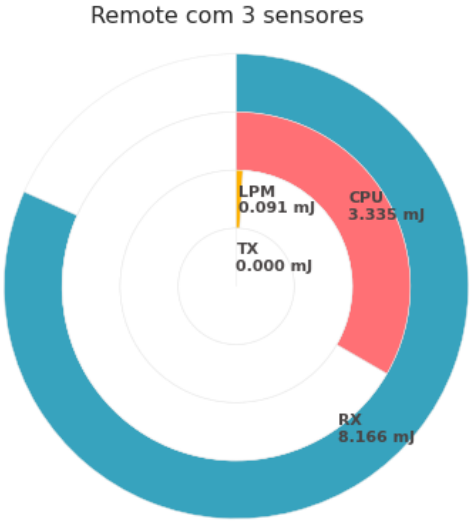
(d) Modelo de ML global usando 9 variáveis preditoras (sensores)

Fonte: Autor

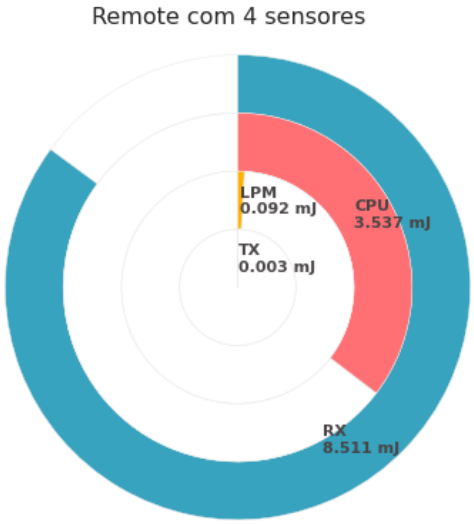
Figura 19 – Consumo de energia do dispositivo Remote para o recebimento da *mensagem 1* para estruturação das variáveis do modelo de ML global



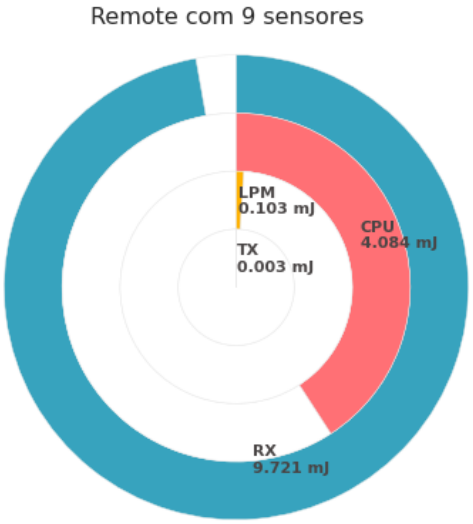
(a) Modelo de ML global usando 2 variáveis preditoras (sensores)



(b) Modelo de ML global usando 3 variáveis preditoras (sensores)



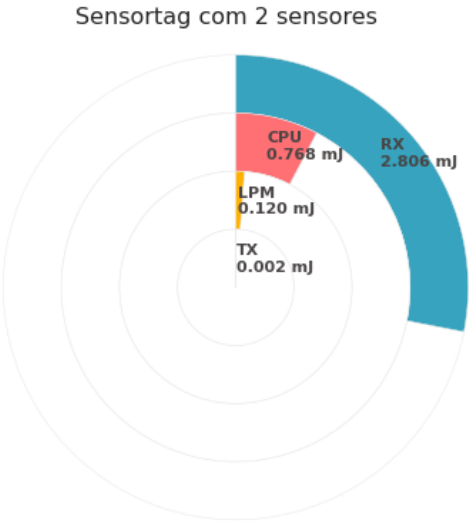
(c) Modelo de ML global usando 4 variáveis preditoras (sensores)



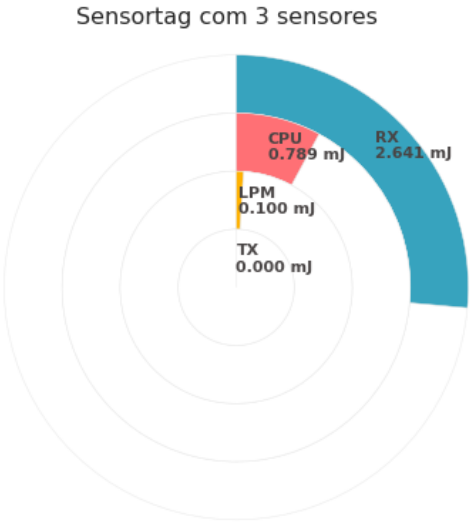
(d) Modelo de ML global usando 9 variáveis preditoras (sensores)

Fonte: Autor

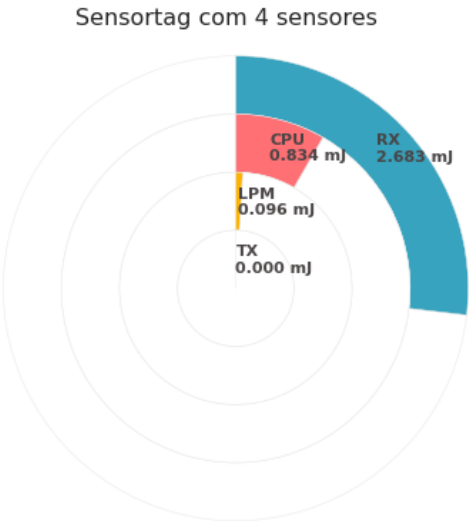
Figura 20 – Consumo de energia do dispositivo Sensortag para o recebimento da *mensagem 1* para estruturação das variáveis do modelo de ML global



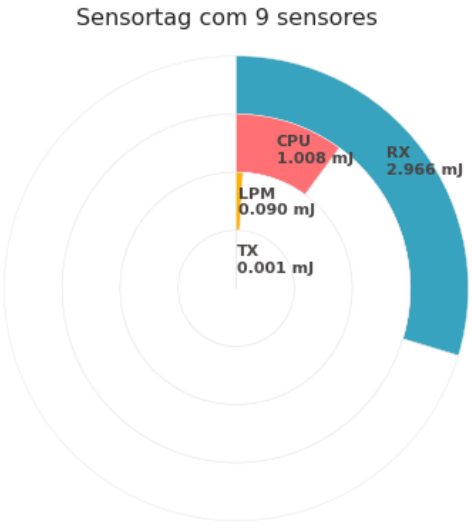
(a) Modelo de ML global usando 2 variáveis preditoras (sensores)



(b) Modelo de ML global usando 3 variáveis preditoras (sensores)



(c) Modelo de ML global usando 4 variáveis preditoras (sensores)



(d) Modelo de ML global usando 9 variáveis preditoras (sensores)

Fonte: Autor

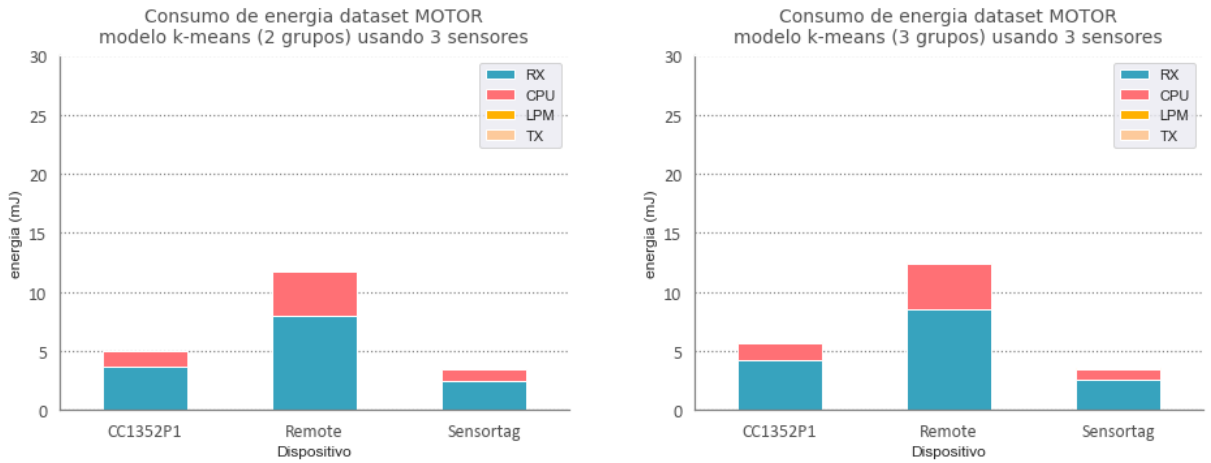
Prosseguindo com o comportamento do consumo de energia dos dispositivos no recebimento dos modelos de ML globais, a seção a seguir apresenta a energia requerida para o recebimento dos modelos de ML.

6.2 Consumo de energia para recebimento do modelo de ML global dispositivos IoT ultra-restritos

Na sequência da mensagem 1, os dispositivos IoT ultra-restritos recebem a mensagem 2, que contém todos os parâmetros do modelo: a frequência das inferências a serem realizadas e os coeficientes (vetores e matrizes de pesos, e viés - para os modelos de regressão). O consumo de energia apresentado a seguir para o recebimento dos modelos de ML globais está segregado nos cenários apresentados no Capítulo anterior.

As Figuras 21 e 22 apresentam o consumo de energia, respectivamente, com o uso do k-means e regressão logística comparativamente em todos os dispositivos avaliados no cenário *Indústria 5.0*.

Figura 21 – Consumo de energia para recebimento do modelo de ML global k-means, cenário *Indústria 5.0* usando 3 sensores



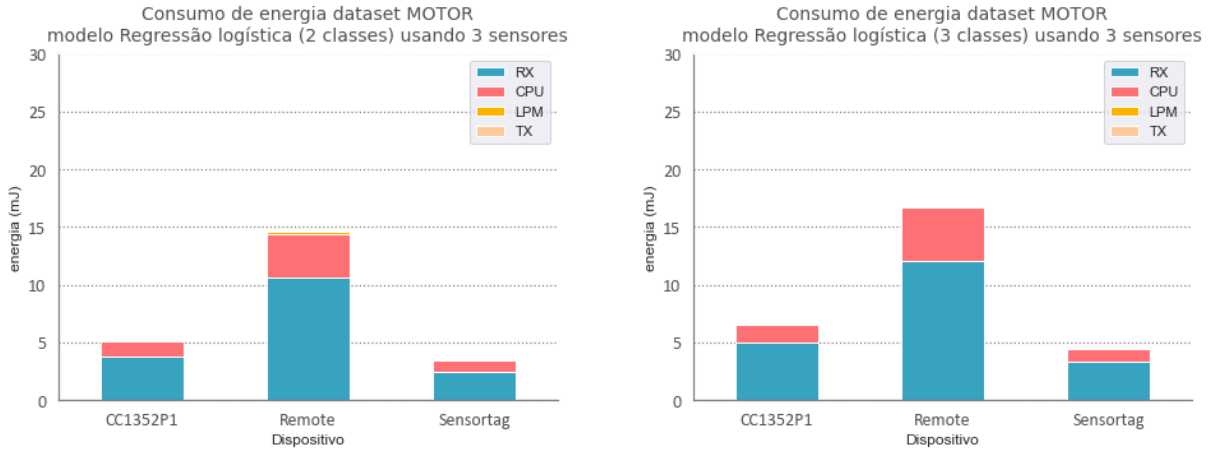
(a) Desfecho do modelo com 2 grupos (b) Desfecho do modelo com 3 sensores

Fonte: Autor

Na Figura 21 observa-se que usar um desfecho do modelo k-means com 3 grupos consome, em média 7,18% mais energia quando comparado ao modelo com 2 grupos (considerando a média referente a cada dispositivo). Ao se analisar por dispositivo, observa-se que o consumo de energia é de 14%, 5.94% e 1.49% maior, respectivamente, para os dispositivos CC1352P1, Remote e Sensortag.

Na Figura 22 observa-se uma diferença ainda maior, com uma média 21% superior para a regressão logística com 3 classes no desfecho do modelo. Por dispositivo, o aumento do consumo de energia é de 30,83%, 15,45% e 31,27%, respectivamente, para CC1352P1,

Figura 22 – Consumo de energia para recebimento do modelo de ML global regressão logística, cenário *Indústria 5.0* usando 3 sensores



(a) Consumo com desfecho do modelo com 2 classes

(b) Consumo com desfecho do modelo com 3 classes

Fonte: Autor

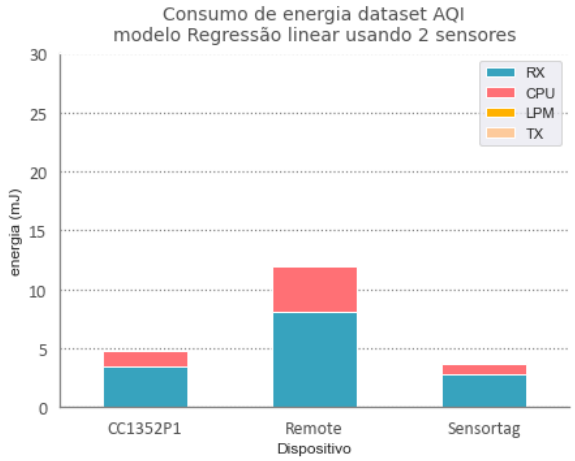
Remote e Sensortag. Essa diferença mais acentuada se evidencia no aumento do RX em 34,68%, 13,87% e 34,66%, e em CPU em 16,15%, 20,21% e 19,02%.

No cenário *cidades inteligentes*, com o experimento *IQAr*, as Figuras 23 (regressão linear), 24, 25, 26 (regressão logística), e, 27, 28 e 29 (k-means) apresentam, o consumo de energia de todos os dispositivos IoT ultra-restritos avaliados.

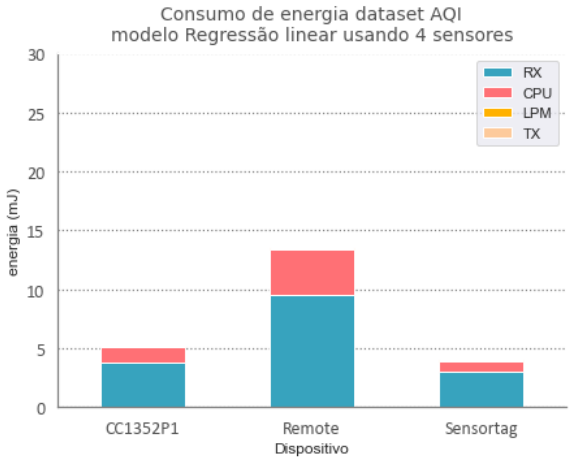
Ao analisar os resultados apresentados na Figura 23 observa-se um incremento gradual do consumo de energia conforme se aumenta o uso do número de sensores como variáveis preditoras do modelo de ML global. O aumento é de 10,43 % e 12,89 %, respectivamente, para o uso de 4 e 9 sensores (em comparação ao uso de 2 sensores).

Com relação ao modelo de ML global regressão logística, os resultados estão segregados por número de variáveis (sensores) utilizadas e número de desfechos. A Figura 24 mostra que o aumento do número de classes no desfecho do modelo de ML global regressão logística usando 2 sensores também resulta no aumento do consumo de energia de todos os dispositivos. Observa-se que o aumento do desfecho de uma solução binária (2 classes) para uma solução multiclasse representa um acréscimo de energia de 7,45%, 8,34% e 8,22%, e, 37,94%, 48,88% e 43,45% respectivamente, para os dispositivos CC1352P1, Remote e Sensortag para 3 e 6 classes de desfecho. Portanto, conforme aumentam-se as possibilidades de desfecho, aumenta-se o consumo de energia. Isso se explica, pois o tamanho da mensagem é maior, fato que exige a recepção de mais bytes pelos dispositivos,

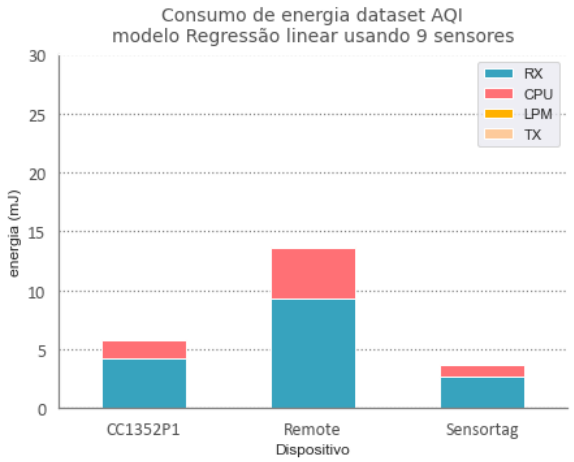
Figura 23 – Consumo de energia para recebimento do modelo de ML global regressão linear, cenário *cidades inteligentes*



(a) Consumo de energia utilizando de 2 variáveis (sensores)



(b) Consumo de energia utilizando 4 variáveis (sensores)



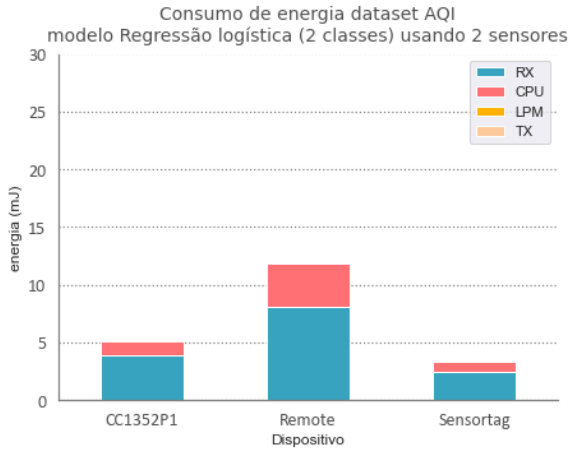
(c) Consumo de energia utilizando 9 variáveis (sensores)

Fonte: Autor

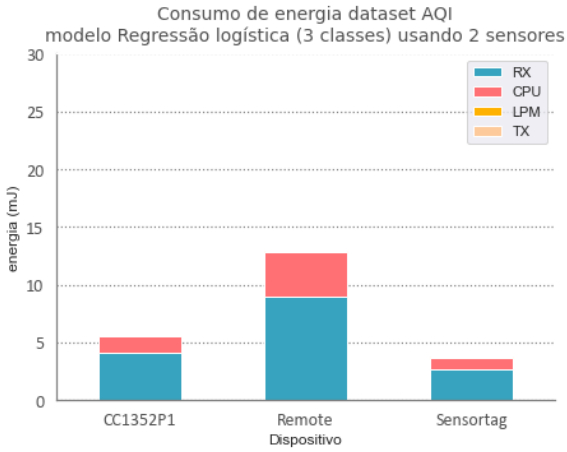
bem como aumenta a necessidade de processamento para a estruturação do modelo de ML global nos dispositivos. Nesse sentido, a avaliação da recepção e processamento dessas mensagens gera um aumento de 7,77%, 10,54% e 8,82% (CC1352P1, Remote e Sensortag: RX) e 28,49%, 29,61% e 29,05% (CC1352P1, Remote e Sensortag: CPU). Dessa forma, mesmo transmitindo-se mais bytes, é mais custoso estruturar o modelo de ML (consumo de processamento) do que receber uma mensagem maior (consumo de recepção da mensagem).

Os resultados do modelo de ML global regressão logística com o uso de 4 e 9

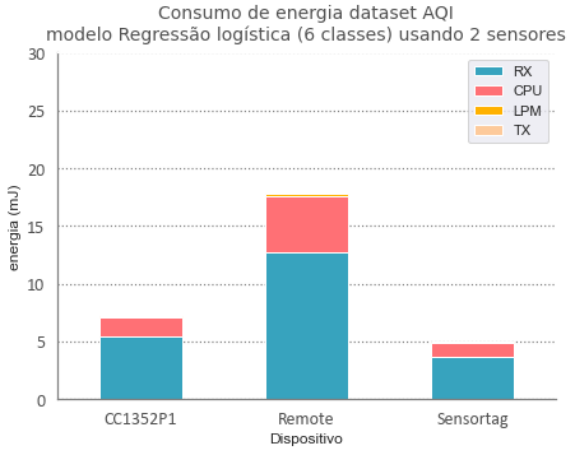
Figura 24 – Consumo de energia para recebimento do modelo de ML global regressão logística usando 2 variáveis (sensores), cenário *cidades inteligentes*



(a) Consumo de energia com desfecho usando 2 classes



(b) Consumo de energia com desfecho usando 3 classes



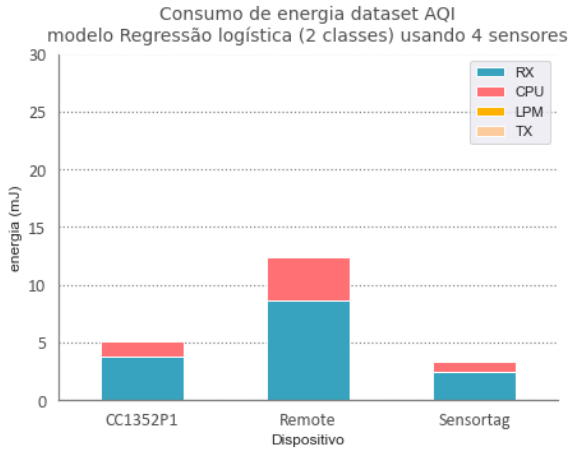
(c) Consumo de energia com desfecho usando 6 classes

Fonte: Autor

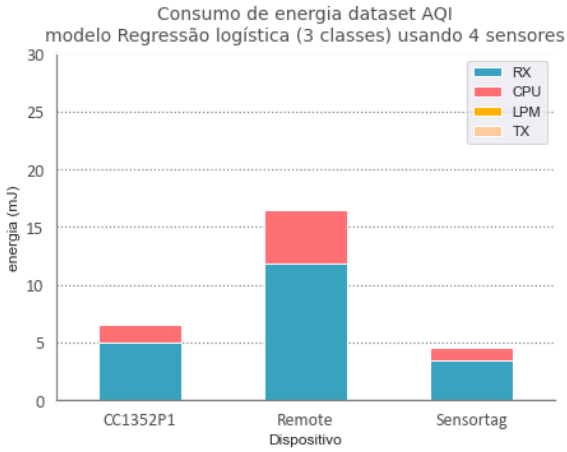
sensores constam nas Figuras 25 e 26.

Os comportamentos referentes ao consumo de energia apresentados nas Figuras 25 e 26, com o uso de 4 e 9 sensores respectivamente, são similares aos observados quando se usam 2 sensores, com um aumento gradual do consumo de energia conforme se aumentam as classes resultantes no desfecho do modelo. O consumo mais acentuado de energia se observa na Figura 26c (uso de 9 sensores e 6 classes no desfecho do modelo). Esse aumento do consumo de energia ocorre tanto com relação às atividades de recepção do modelo de

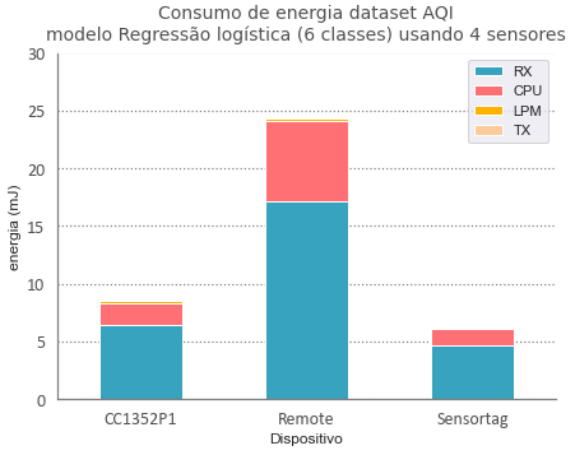
Figura 25 – Consumo de energia para recebimento do modelo de ML global regressão logística usando 4 variáveis (sensores), cenário *cidades inteligentes*



(a) Consumo de energia com desfecho usando 2 classes



(b) Consumo de energia com desfecho usando 3 classes



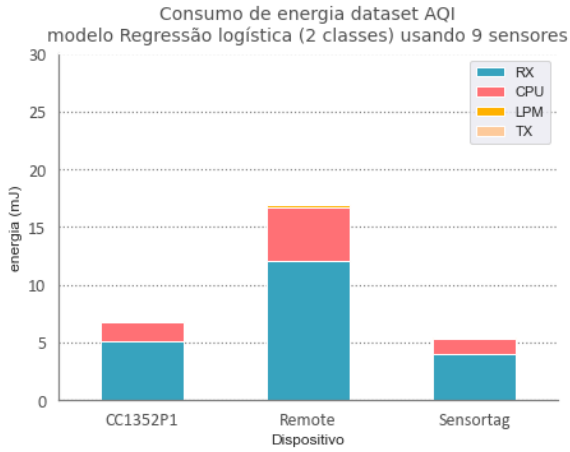
(c) Consumo de energia com desfecho usando 6 classes

Fonte: Autor

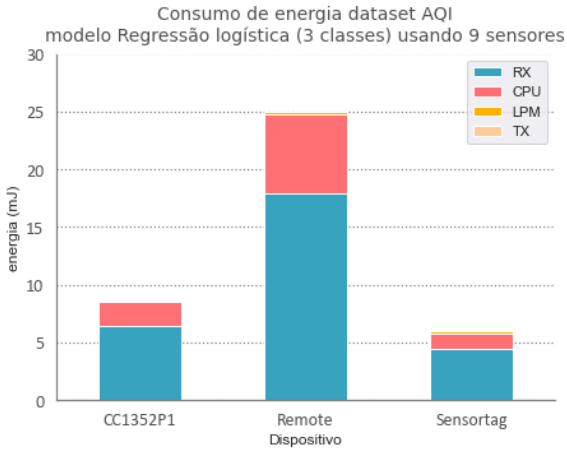
ML global (RX) quanto no tempo gasto para estruturação do modelo no dispositivo IoT (CPU).

Comparando-se o emprego de um desfecho binário (2 classes) nas Figuras 24a e 25a, o consumo de energia utilizando 2 ou 4 sensores como variáveis preditoras é similar (o percentual de aumento observado não ultrapassa 5% no dispositivo Remote). Entretanto, o mesmo desfecho quando comparado ao uso de 9 sensores (Figura 26a) aumenta o consumo de energia em 32,55%, 49,72% e 60,13% (para recepção) e 27%, 21,91% e 50,97% (para

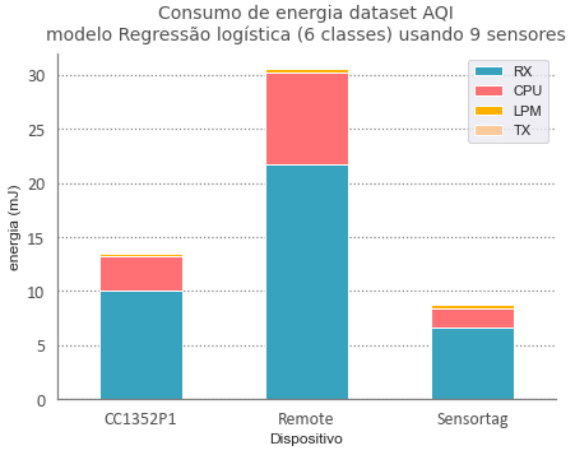
Figura 26 – Consumo de energia para recebimento do modelo de ML global regressão logística usando 9 variáveis (sensores), cenário *cidades inteligentes*



(a) Consumo de energia com desfecho usando 2 classes



(b) Consumo de energia com desfecho usando 3 classes



(c) Consumo de energia com desfecho usando 6 classes

Fonte: Autor

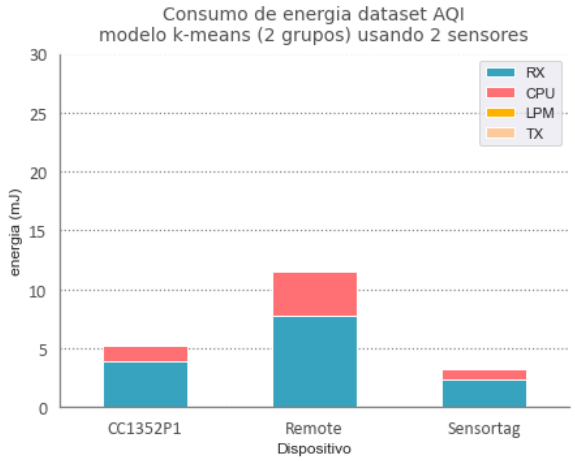
processamento), respectivamente para os dispositivos CC1352P1, Remote e Sensortag.

Essa diferença ocorre em virtude da característica do modelo de ML regressão logística apresentado na Eq. 4.3, pois o número de coeficientes do modelo depende do número de variáveis (sensores) em combinação com o número de desfechos. A matriz de interceptos corresponde ao desfecho, mantendo-se uma matriz 1x2 independentemente do número de variáveis, enquanto a matriz de pesos varia conforme o número de sensores utilizados no modelo de ML. No caso concreto do desfecho binário, a Figura 24a (2 sensores)

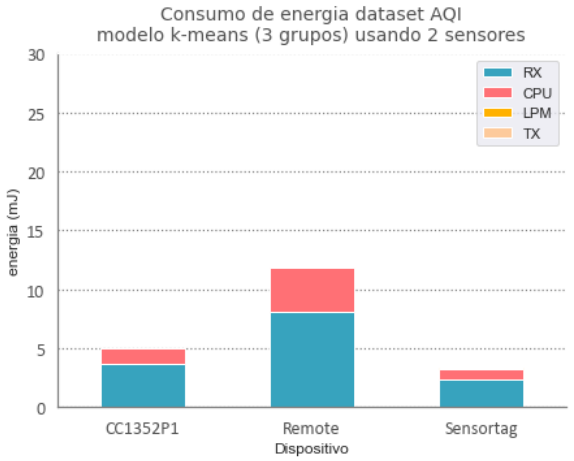
resulta em uma matriz de pesos 2x2, enquanto na Figura 25a (4 sensores) e 26a (9 sensores) as matrizes são 4x2 e 9x2 respectivamente. O incremento de Bytes com a utilização de 4 sensores é de 59%, enquanto com a utilização de 9 sensores o incremento é de mais de 400%. Por esse motivo observa-se o aumento no consumo de energia, que incide tanto no tempo para receber o modelo quanto para estruturar o modelo de ML no dispositivo IoT ultra-restrito.

O próximo modelo a ser analisado é o k-means, cujos resultados constam nas Figuras 27, 28 e 29.

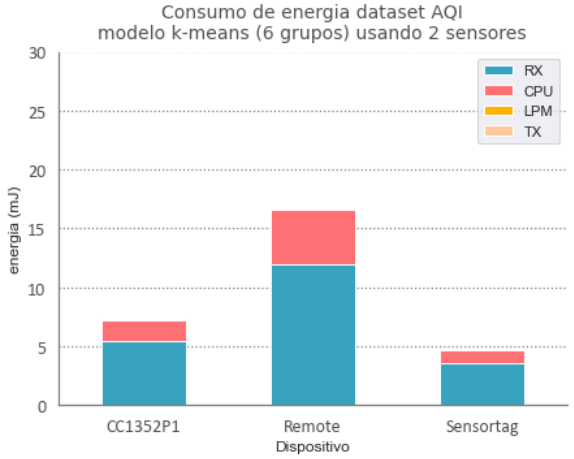
Figura 27 – Consumo de energia para recebimento do modelo de ML global k-means usando 2 variáveis (sensores), cenário *cidades inteligentes*



(a) Consumo de energia com desfecho usando 2 grupos



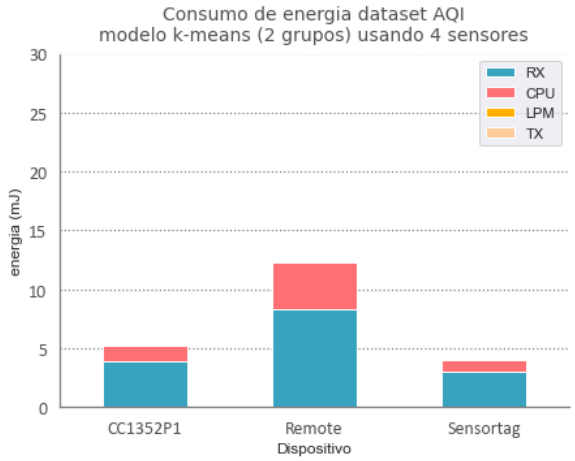
(b) Consumo de energia com desfecho usando 3 grupos



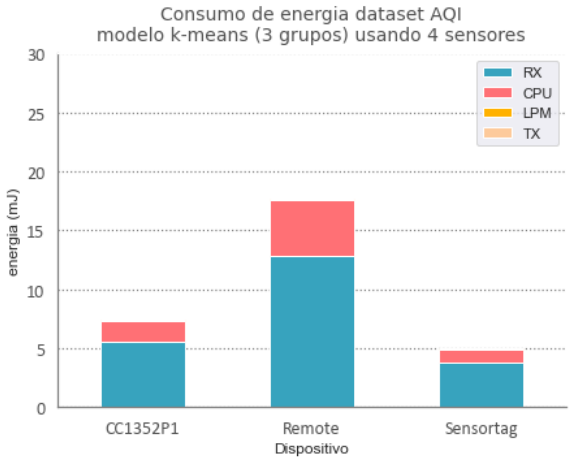
(c) Consumo de energia com desfecho usando 6 grupos

Fonte: Autor

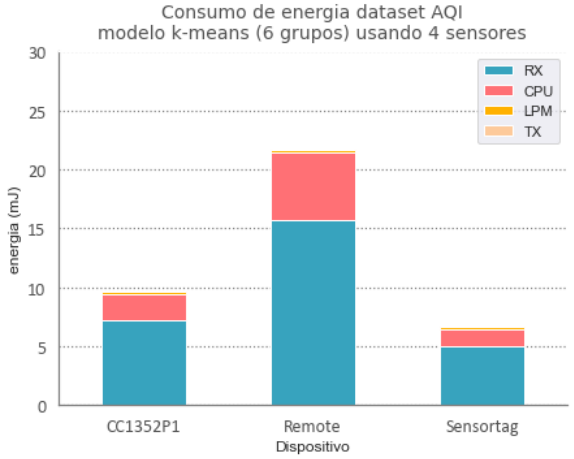
Figura 28 – Consumo de energia para recebimento do modelo de ML global k-means usando 4 variáveis (sensores), cenário *cidades inteligentes*



(a) Consumo de energia com desfecho usando 2 grupos



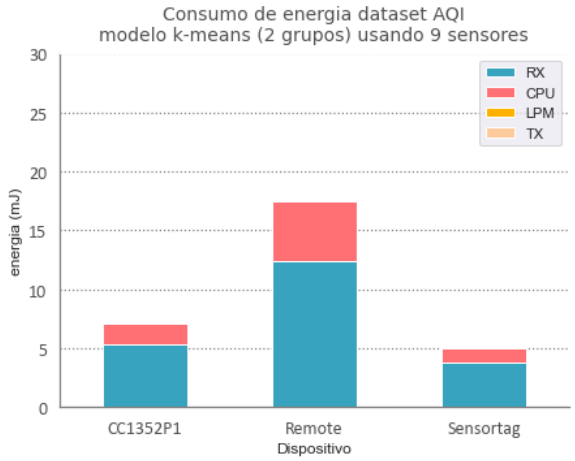
(b) Consumo de energia com desfecho usando 3 grupos



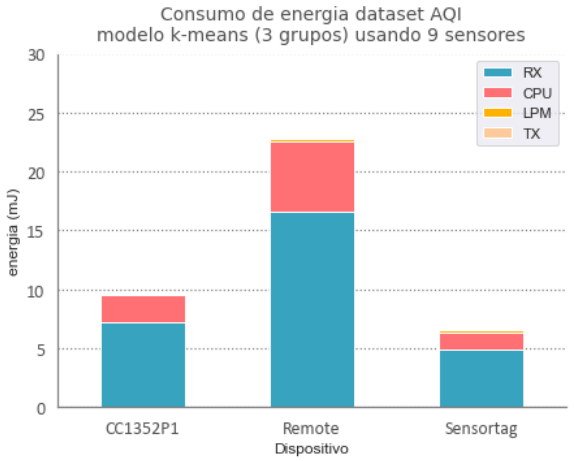
(c) Consumo de energia com desfecho usando 6 grupos

Fonte: Autor

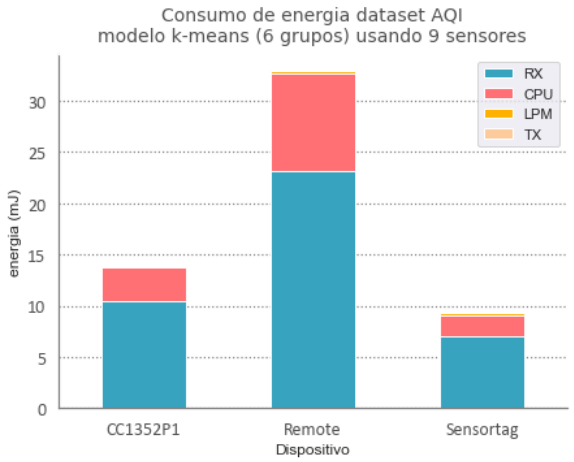
Figura 29 – Consumo de energia para recebimento do modelo de ML global k-means usando 9 variáveis (sensores), cenário *cidades inteligentes*



(a) Consumo de energia com desfecho usando 2 grupos



(b) Consumo de energia com desfecho usando 3 grupos



(c) Consumo de energia com desfecho usando 6 grupos

Fonte: Autor

Assim como o observado nos resultados referentes ao modelo regressão logística, no modelo k-means, conforme se aumentam os grupos de desfecho e o número de sensores como variáveis preditoras, há um aumento no consumo de energia, em percentuais similares.

Uma vez recebidos os modelos de ML globais, os dispositivos passam a tomar decisões baseadas na aplicação dos respectivos modelos de ML. Os dispositivos IoT ultra-restritos executam a seguinte sequência de ações para a tomada de decisão após a conclusão

do treinamento federado:

1. Recebem a *mensagem 1* do participante contendo os sensores a serem utilizados e o intervalo de tempo de coleta dos dados dos sensores e realização da inferência;

2. Recebem a *mensagem 2* do participante contendo os parâmetros (coeficientes) do modelo de ML global e (a) a classe ou grupo que deflagrará a ação a ser tomada (no caso dos modelos de classificação ou agrupamento, respectivamente), ou (b) o valor máximo aceitável que, se ultrapassado, deflagra a ação (no caso de modelos de regressão linear);

3. Realizam a inferência no intervalo de tempo previsto na mensagem 2.

A seção a seguir apresenta os resultados referentes ao consumo de energia para a realização da inferência pelos dispositivos.

6.3 Consumo de energia para a realização de inferência pelos dispositivos IoT ultra-restritos

A realização da inferência pelos dispositivos IoT ultra-restritos é o recurso que viabiliza a tomada de decisão inteligente e autônoma baseada nos dados coletados dos sensores e que passam pelo modelo de ML global recebido pelo participante do *framework* FedSensor.

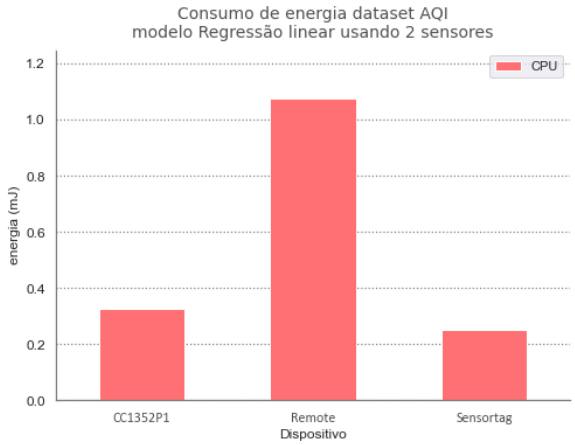
A inferência realizada nos dispositivos IoT ultra-restritos considera exclusivamente o consumo de energia referente ao processamento (CPU), uma vez que as demais atividades do dispositivo IoT ultra-restritos (LPM, RX, TX) não são realizadas durante a inferência. As Figuras 30, 31, 32 apresentam o consumo de energia para realização de inferência nos cenários IoT propostos.

A Figura 30 apresenta o consumo de energia para a realização de inferência usando o modelo de ML global regressão linear, considerando o uso de 2, 4 ou 9 sensores.

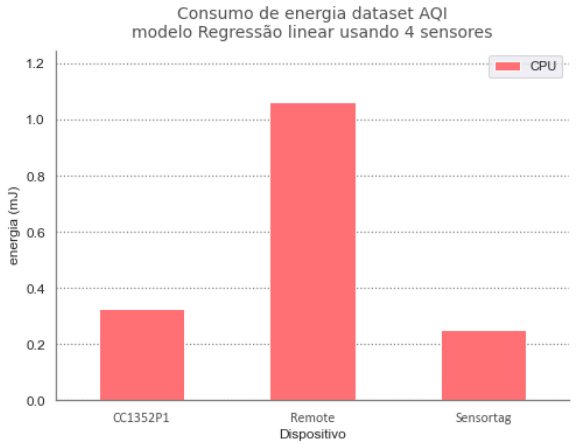
Ao analisar o consumo de energia apresentado na Figura 30, conclui-se que a energia requerida para inferência é o mesmo independentemente do número de sensores utilizados como variáveis preditoras do modelo de ML global regressão linear. Nesse caso, o desfecho do modelo de ML global do FedSensor é um valor contínuo, que será comparado com o

valor máximo aceitável recebido na *mensagem 2* para tomada de decisão autônoma pelo dispositivo.

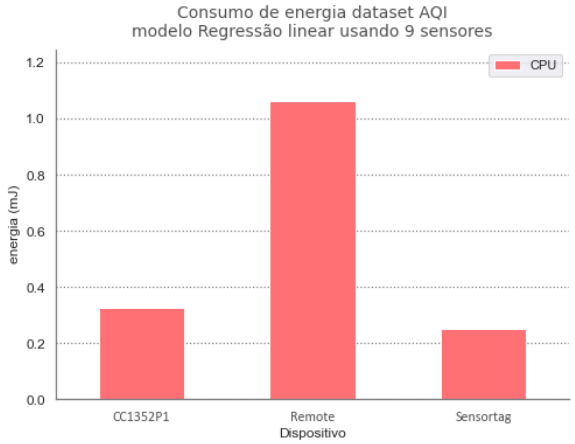
Figura 30 – Consumo de energia para a realização de inferência pelos dispositivos IoT ultra-restritos usando o modelo de ML global regressão linear



(a) Consumo de energia utilizando de 2 variáveis (sensores)



(b) Consumo de energia utilizando 4 variáveis (sensores)



(c) Consumo de energia utilizando 9 variáveis (sensores)

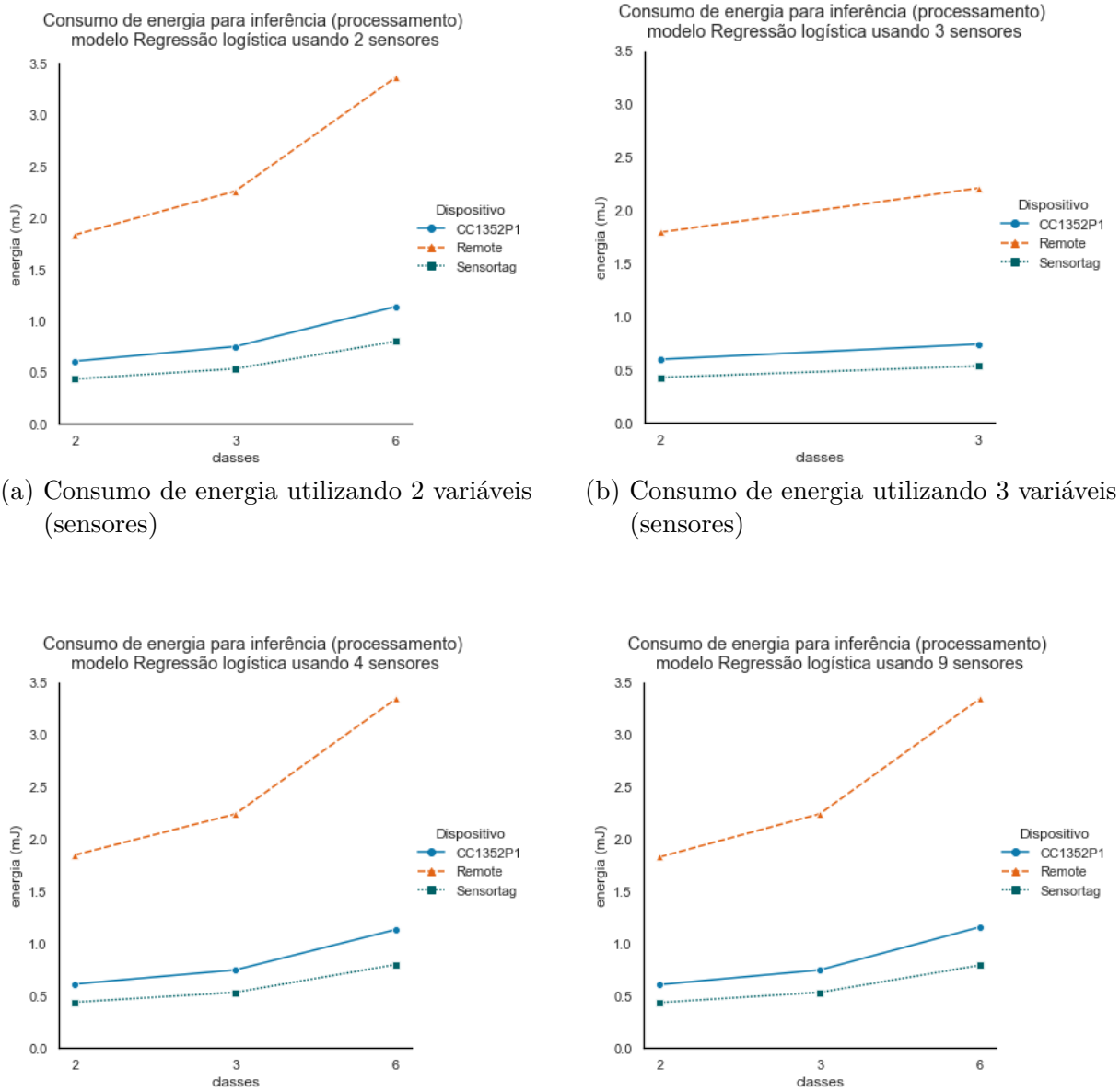
Fonte: Autor

A seguir, apresentam-se nas Figuras 31 e 32 o consumo de energia para realização de inferência aplicando, respectivamente, o modelo de ML global regressão logística e k-means, em aplicações IoT com 2, 3, 4 e 9 sensores como variáveis preditoras.

Os valores apresentados consideram a média do volume total de inferências realizados. No caso da inferência usando 3 sensores, tanto para a regressão logística quanto para k-means, considera-se apenas o experimento do cenário *Indústria 5.0*. No caso da

inferência usando 4 e 9 sensores, também para regressão logística e k-means, considera-se apenas o experimento do cenário *cidades inteligentes*.

Figura 31 – Consumo de energia para a realização de inferência pelos dispositivos IoT ultra-restritos usando o modelo de ML global regressão logística



(a) Consumo de energia utilizando 2 variáveis (sensores)

(b) Consumo de energia utilizando 3 variáveis (sensores)

(c) Consumo de energia utilizando 4 variáveis (sensores)

(d) Consumo de energia utilizando 9 variáveis (sensores)

Fonte: Autor

A análise do perfil de consumo de energia do modelo de ML global regressão logística apresentado na Figura 31 permite observar:

- Com relação ao uso de 2 sensores e 2 classes, há um aumento médio do consumo de

energia:

  - De 18,80% com o uso de 3 classes;
  - De 45,88% com o uso de 6 classes.

- Com relação ao uso de 3 sensores e 2 classes, há um aumento médio do consumo de energia:

  - De 19,41% com o uso de 3 classes;

- Com relação ao uso de 4 sensores e 2 classes, há um aumento médio do consumo de energia:

  - De 17,70% com o uso de 3 classes;
  - De 45,23% com o uso de 6 classes;

- Com relação ao uso de 6 sensores e 2 classes, há um aumento médio do consumo de energia:

  - De 18,45% com o uso de 3 classes;
  - De 45,96% com o uso de 6 classes.

Conclui-se, portanto, que o consumo de energia mantém o perfil observado quando do recebimento do modelo de ML global pelos dispositivos, indicando que o aumento do número de desfechos ocasiona um aumento no consumo de energia dos dispositivos IoT ultra-restritos para a realização de inferência.

A Figura 32 apresenta o perfil de consumo de energia dos dispositivos IoT aplicando o modelo de ML global k-means. Obtém-se os seguintes resultados:

- Com relação ao uso de 2 sensores e 2 classes, há um aumento médio do consumo de energia:

  - De 16,81% com o uso de 3 classes;
  - De 38,14% com o uso de 6 classes;

- Com relação ao uso de 3 sensores e 2 classes, há um aumento médio do consumo de energia:

  - De 12,28% com o uso de 3 classes;

- Com relação ao uso de 4 sensores e 2 classes, há um aumento médio do consumo de energia:
  - De 12,14% com o uso de 3 classes;
  - De 35,10% com o uso de 6 classes;
- Com relação ao uso de 6 sensores e 2 classes, há um aumento médio do consumo de energia:
  - De 17,98% com o uso de 3 classes;
  - De 38,51% com o uso de 6 classes.

Ao se comparar o gasto energético para inferência do modelo regressão logística com o modelo k-means, observa-se reduzido percentual de aumento do consumo de energia no modelo k-means. Isso se deve ao fato da operação realizada para inferência ser menos custosa para o k-means (que se baseia na distância euclidiana), quando comparada à regressão logística, que requer um cálculo mais complexo.

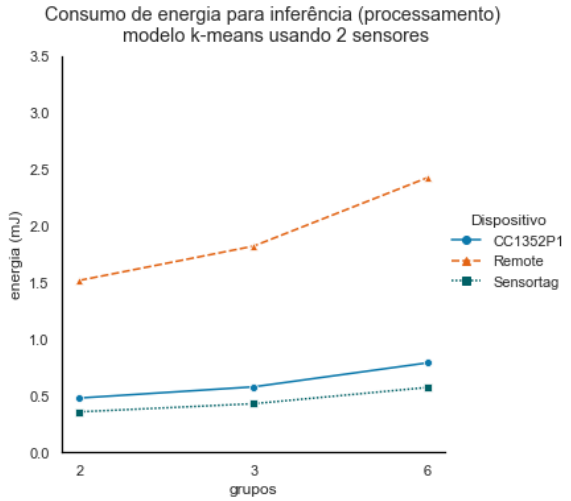
As seções apresentadas até o momento apresentam o consumo de energia do FedSensor referente aos dispositivos IoT ultra-restritos. A seção a seguir apresenta um comparativo entre a recepção dos modelos de ML globais e a realização de inferências.

6.4 Comparativo entre a atualização do modelo global e a realização de inferências

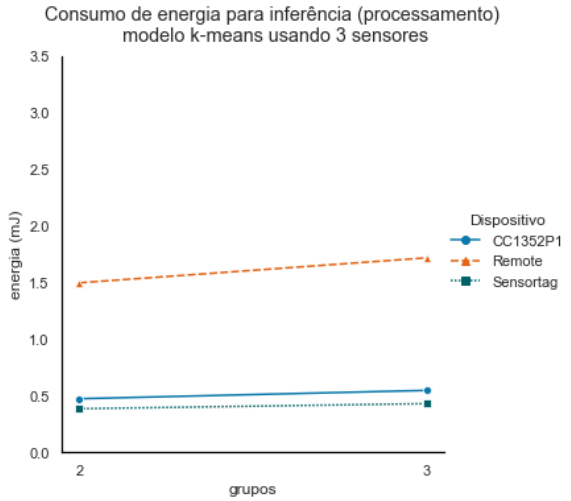
Uma vez apresentados os resultados do consumo de energia das principais ações a serem realizadas pelos dispositivos IoT ultra-restritos (recepção e estruturação do modelo de ML global, e realização de inferências), é necessário avaliar o consumo total de energia e a vida útil das baterias dos dispositivos para verificar a hipótese de utilização desses dispositivos em uma arquitetura de FL com o uso do FedSensor.

Nesse sentido, avalia-se comparativamente o consumo de energia referente à recepção do modelo de ML global e à realização de inferências. Para isso, realiza-se a atualização do modelo de ML global em diferentes intervalos de tempo, a cada: 7,5, 15 e 30 minutos, e 1, 2, 4, 8 e 24 horas. Esses intervalos de tempo avaliados correspondem ao tempo envolvido para a conclusão de um treinamento federado entre o gerenciador e os participantes envolvidos

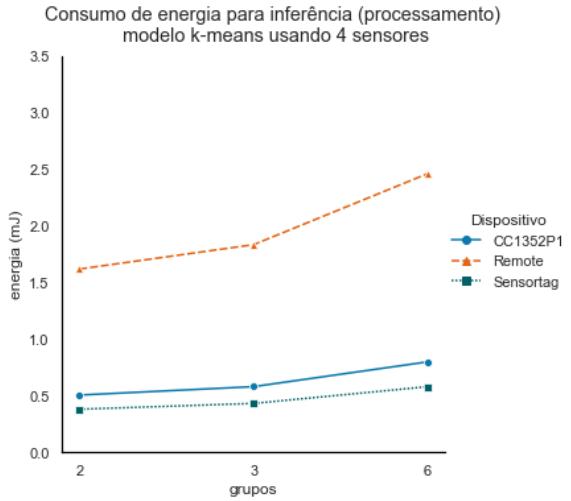
Figura 32 – Consumo de energia para a realização de inferência pelos dispositivos IoT ultra-restritos usando o modelo de ML global k-means



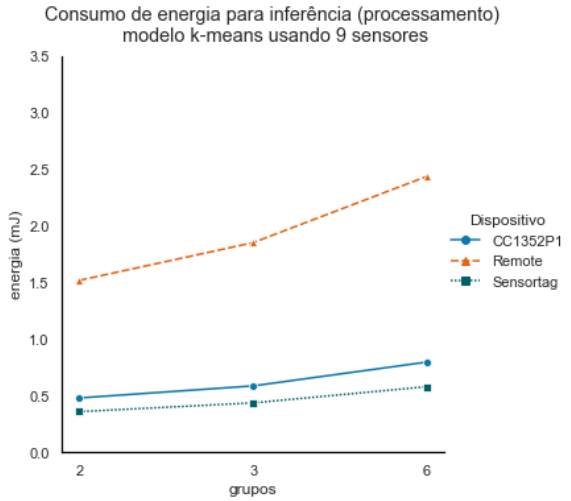
(a) Consumo de energia utilizando 2 variáveis (sensores)



(b) Consumo de energia utilizando 3 variáveis (sensores)



(c) Consumo de energia utilizando 4 variáveis (sensores)



(d) Consumo de energia utilizando 9 variáveis (sensores)

Fonte: Autor

e, após a conclusão do treinamento, ocorre o envio do modelo de ML global para os dispositivos IoT ultra-restritos participantes daquela aplicação IoT.

As Figuras 33 e 34 apresentam o comparativo da energia média gasta para a atualização dos modelos de ML globais e realização de inferências, segregadas por dispositivo

nos intervalos de tempo apresentados. Nessas figuras não consta a energia envolvida referente ao envio da *mensagem 1*, uma vez que se refere à alteração dos sensores envolvidos no modelo de ML global - isso se deve ao fato de que as mudanças de variáveis no modelo de ML global são menos constantes do que as iterações de treinamento federado. Essas Figuras apresentam, adicionalmente, apresentam a vida útil da bateria dos dispositivos IoT ultra-restritos com a utilização do FedSensor.

A Figura 33 apresenta o comparativo considerando 30 segundos para realização de inferências e a Figura 34, 10 segundos.

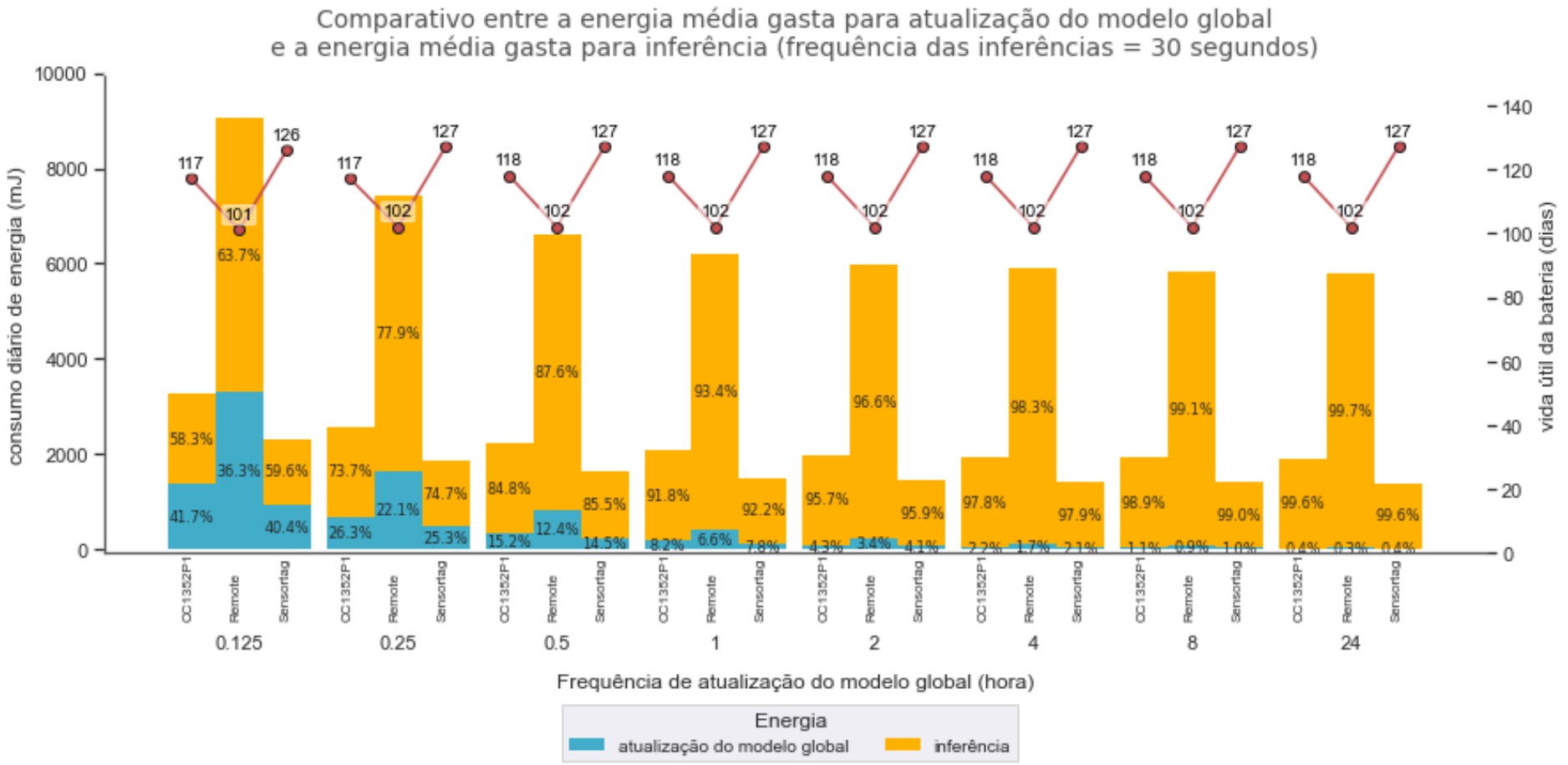
As Figuras 33 e 34 mostram que, apesar do consumo individual de energia para a realização de inferências ser menor quando comparado ao consumo individual para atualização do modelo de ML global, o volume de inferências realizadas nos dispositivos IoT ultra-restritos guia o consumo de energia. Portanto, pode-se concluir que não é o volume de iterações de um treinamento federado que aumenta o consumo de energia nos dispositivos. Por outro lado, mesmo uma frequência de geração de modelos de ML globais a cada 7,5 minutos (o que se mostra um intervalo de tempo que não se observa na prática, pois exigiria que o treinamento do modelo ocorresse 192 vezes no mesmo dia) não impacta na vida útil dos dispositivos.

Esse comportamento é o mesmo, independentemente do intervalo de realização de inferências pelos dispositivos, com uma média de 2 dias a menos de vida útil de baterias com inferências realizadas a cada 10 segundos (em comparação às inferências realizadas a cada 30 segundos).

Outro fator fundamental a ser observado e levantado nas hipóteses, é a vida útil das baterias dos dispositivos IoT ultra-restritos, pois, independentemente se uma aplicação IoT é uma solução implantada e operacional nesses dispositivos, a vida útil das baterias deve ser levada em consideração. Isso porque os dispositivos IoT ultra-restritos normalmente são implantados em locais sem assistência humana, e precisam operar por dias ou meses sem que seja necessário substituir as baterias. Nesse sentido, uma aplicação IoT que consuma excessivamente as baterias, e não mantenha o consumo de energia dentro da expectativa de dias ou meses, não é uma aplicação viável.

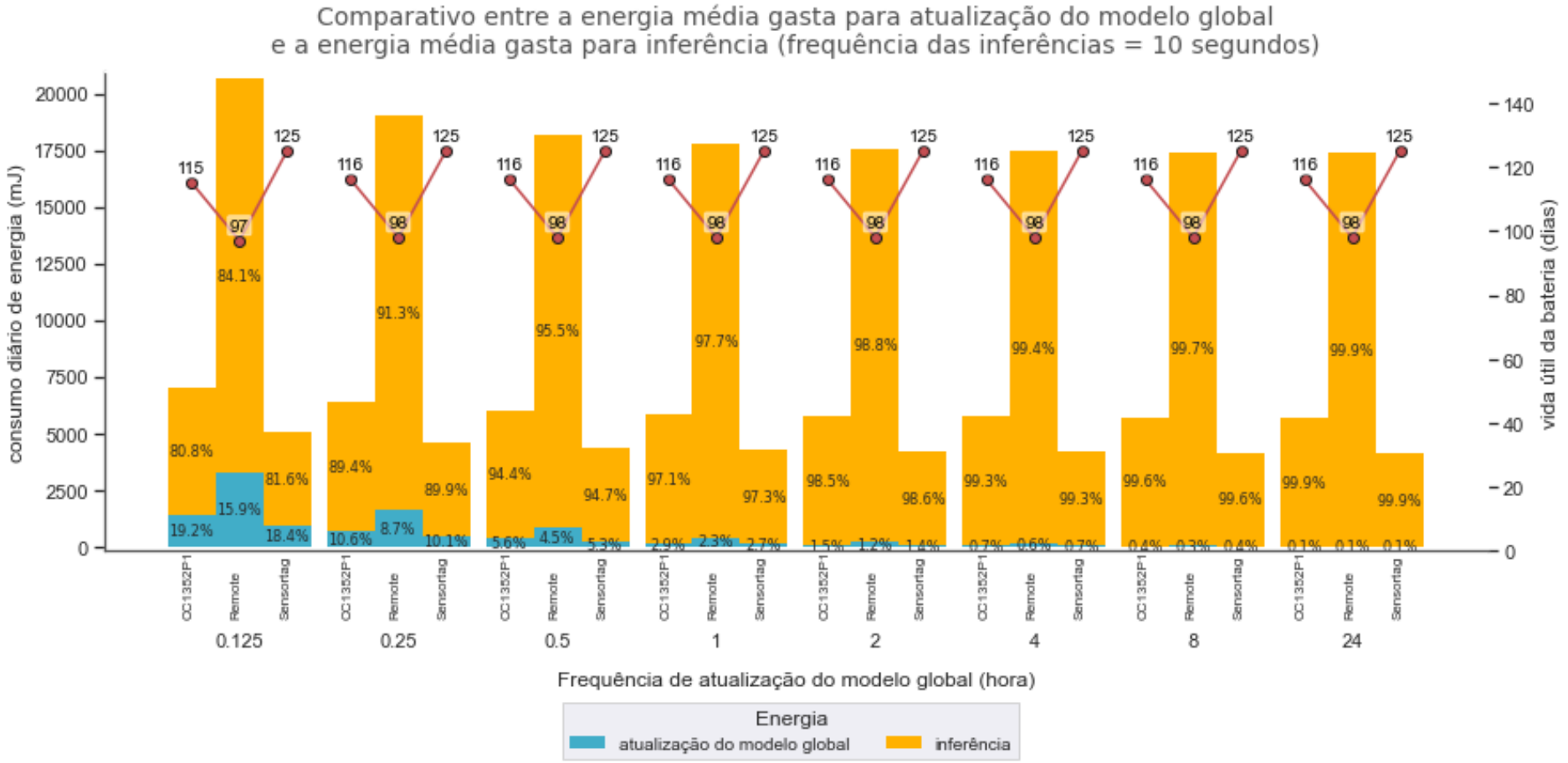
Por isso, as Figuras 35, 36, 37 e 38 apresentam a vida útil das baterias dos dispositivos CC1352P1, Remote e Sensortag em diferentes condições de operação do

Figura 33 – Comparativo do consumo de energia nas principais ações do FedSensor: atualização do modelo de ML global e realização de inferências (30 segundos)



Fonte: Autor

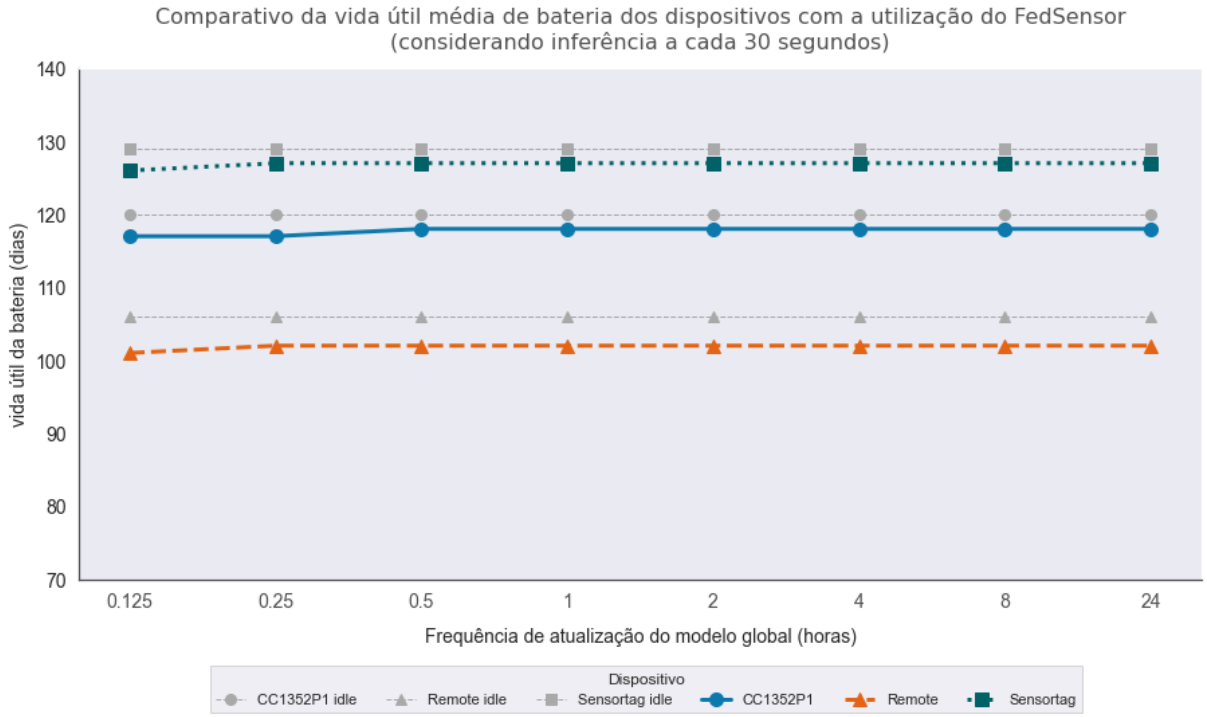
Figura 34 – Comparativo do consumo de energia nas principais ações do FedSensor: atualização do modelo de ML global e realização de inferências (10 segundos)



Fonte: Autor

FedSensor.

Figura 35 – Comparativo da vida útil média de bateria dos dispositivos (em dias), considerando variação da frequência de atualização do modelo de ML global (em horas), com inferências a cada 30 segundos



Fonte: Autor

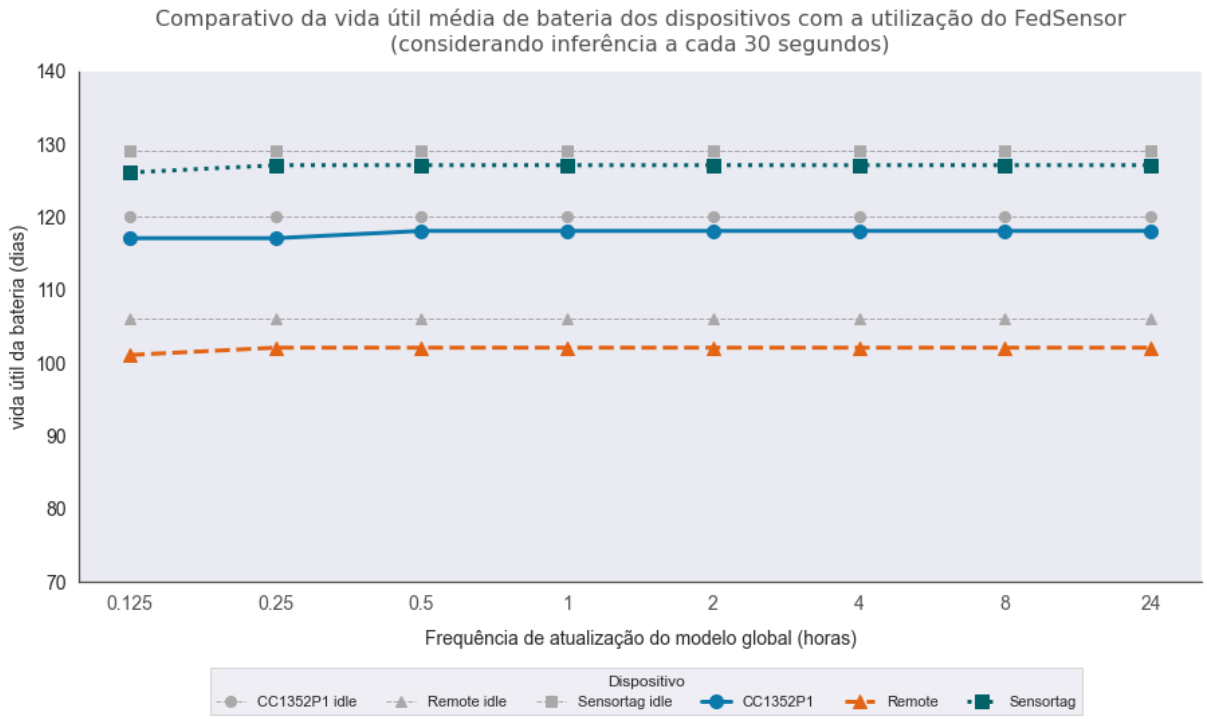
A Figura 35 apresenta a vida útil das baterias dos dispositivos considerando os oito intervalos de tempo apresentados para atualização do modelo de ML global (em horas), com a inferência realizada a cada 30 segundos.

Na Figura 35, a vida útil das baterias dos dispositivos IoT ultra-restritos avaliados fica próxima da vida útil máxima das baterias. A obtenção da vida útil máxima das baterias ocorre com a avaliação dos dispositivos IoT no formato ligado, mas sem atividade de recepção de modelos de ML, nem a realização de inferências. Nesse sentido, conclui-se que a realização de inferências a cada 30 segundos reduz a vida útil em 1,87%, 3,88% e 1,64%, respectivamente, para os dispositivos CC1352P1, Remote e Sensortag. Esse reduzido percentual mostra que os dispositivos IoT ultra-restritos são capazes de participar de uma arquitetura federada usando o FedSensor. Esse percentual se mantém o mesmo independentemente da frequência de atualização do modelo global, mesmo em um cenário com treinamentos federados ocorrendo a cada 7,5 minutos (e que exigem dos dispositivos

IoT a atualização do modelo global 192 vezes por dia).

Além disso, avalia-se o FedSensor em cenários que exigem mais inferências, conforme mostra a Figura 36, com inferências realizadas a cada 10 segundos.

Figura 36 – Comparativo da vida útil média de bateria dos dispositivos (em dias), considerando variação da frequência de atualização do modelo de ML global (em horas), com inferências a cada 10 segundos



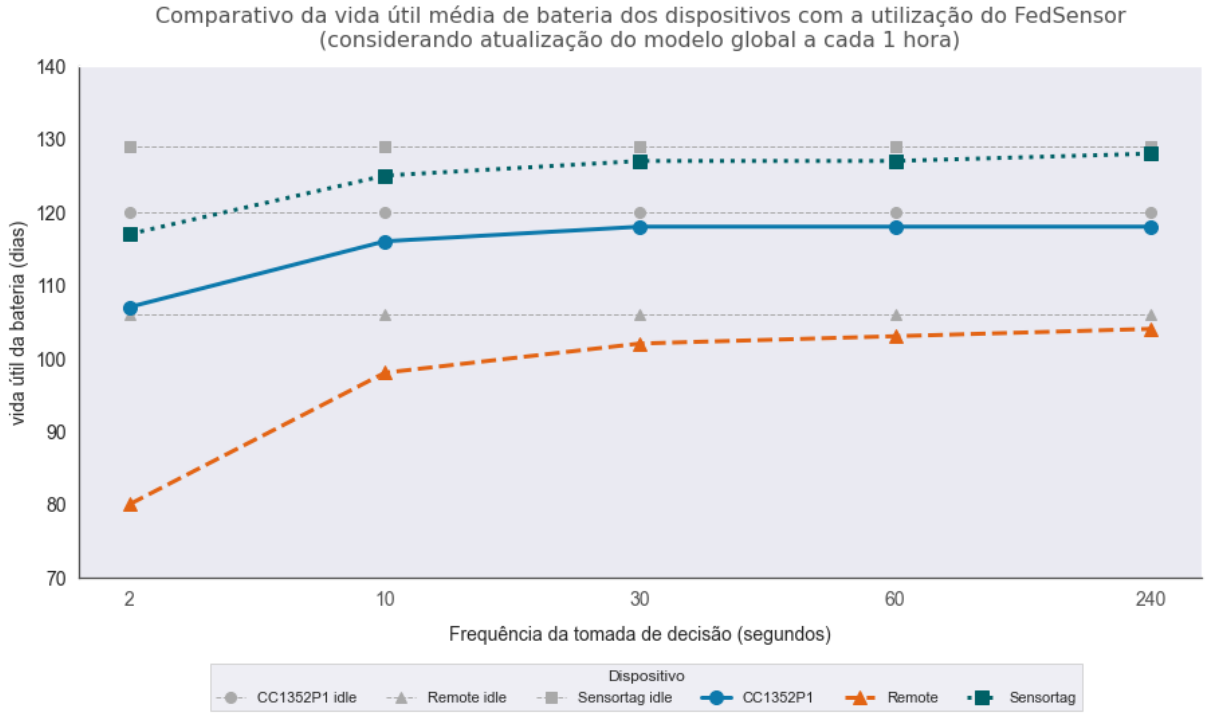
Fonte: Autor

Mesmo em um cenário que exige mais recursos de processamento do dispositivo, como é o apresentado na Figura 36, os percentuais de redução da vida útil das baterias são de 3,43%, 7,66% e 3,10%, respectivamente, para os dispositivos CC1352P1, Remote e Sensortag. Portanto, conclui-se que o FedSensor é viável para os dispositivos IoT ultra-restritos participarem de uma arquitetura de FL, realizando a atualização dos modelos de ML global (sem que seja necessário implantar novo firmware - e indisponibilizando o dispositivo, por exemplo) e tomando decisões autônomas, resultante de modelos de ML executados nos próprios dispositivos.

Para ampliar os resultados referentes à vida útil, a Figura 37 mostra a vida útil das baterias dos dispositivos IoT avaliados, considerando a atualização do modelo de ML global a cada uma hora nas seguintes frequências de realização de inferência (em segundos):

2, 10, 30, 60 e 240.

Figura 37 – Comparativo da vida útil média de bateria dos dispositivos (em dias), considerando variação da frequência de realização de inferências (tomada de decisão) pelo dispositivo (em segundos), com atualizações do modelo de ML global a cada uma hora



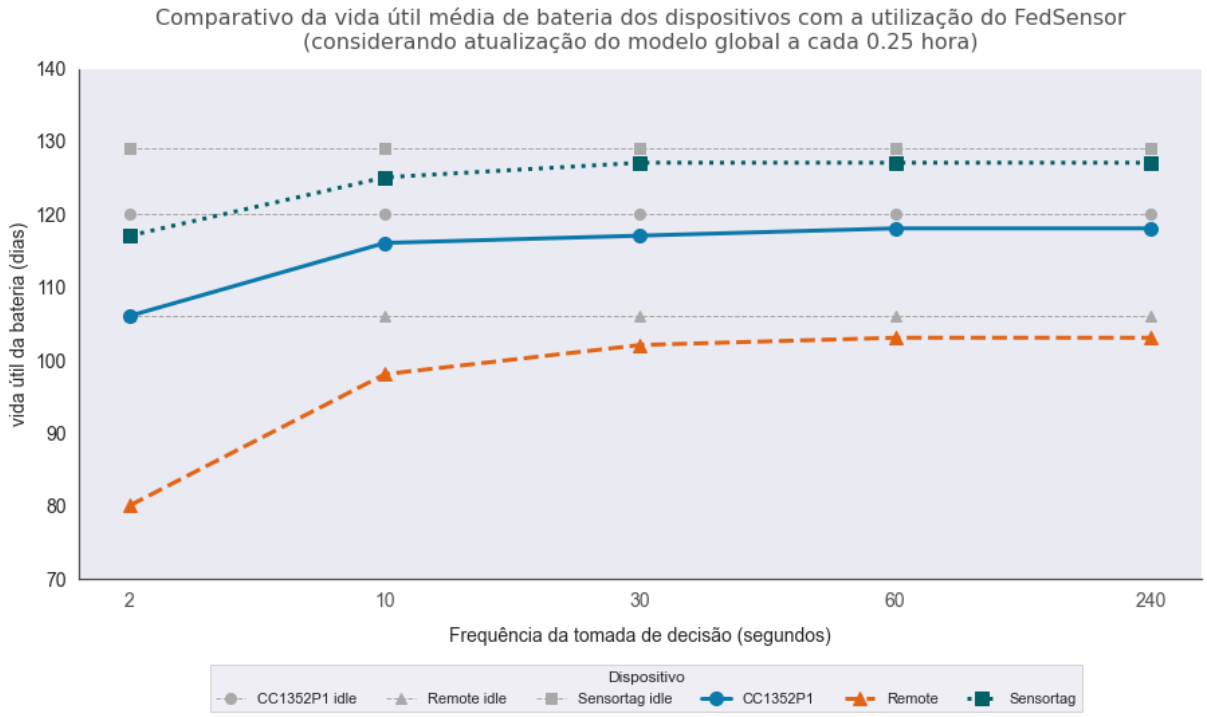
Fonte: Autor

Figura 37 evidencia que a realização de inferências a cada dois segundos reduz de maneira mais severa a bateria dos dispositivos IoT avaliados, em percentuais de 10,83%, 24,53% e 9,30%, respectivamente, para os dispositivos CC1352P1, Remote e Sensortag - mas mesmo havendo redução esse percentual viabiliza a utilização desses dispositivos em arquiteturas de FL, como é o caso do FedSensor.

O aumento da frequência de atualização do modelo de ML global para o maior volume de troca de modelos de ML globais avaliado nos experimentos (7,5 minutos, com 192 alterações diárias) não traz diferenças no percentual de redução da vida útil da bateria, conforme se observa na Figura 38.

Todo o comportamento referente ao consumo de energia apresentado confirma a hipótese de viabilidade de utilização de dispositivos IoT ultra-restritos em uma arquitetura de FL como o FedSensor, pois mesmo em um cenário que exige muitas inferências (uma a

Figura 38 – Comparativo da vida útil média de bateria dos dispositivos (em dias), considerando variação da frequência de realização de inferências (tomada de decisão) pelo dispositivo (em segundos), com atualizações do modelo de ML global a cada 7,5 minutos



Fonte: Autor

cada dois segundos), o consumo de energia ainda permite meses de utilização do dispositivo.

Dentro do contexto de consumo de energia, está a seleção de variáveis, que tem por objetivo encontrar o melhor conjunto de variáveis (no caso do FedSensor, sensores do dispositivo) que resultam no aumento da acurácia do modelo. Nesse sentido, a seção a seguir apresenta um comparativo com o envolvimento da seleção de variáveis.

6.5 Seleção de variáveis e o relacionamento com o consumo de energia

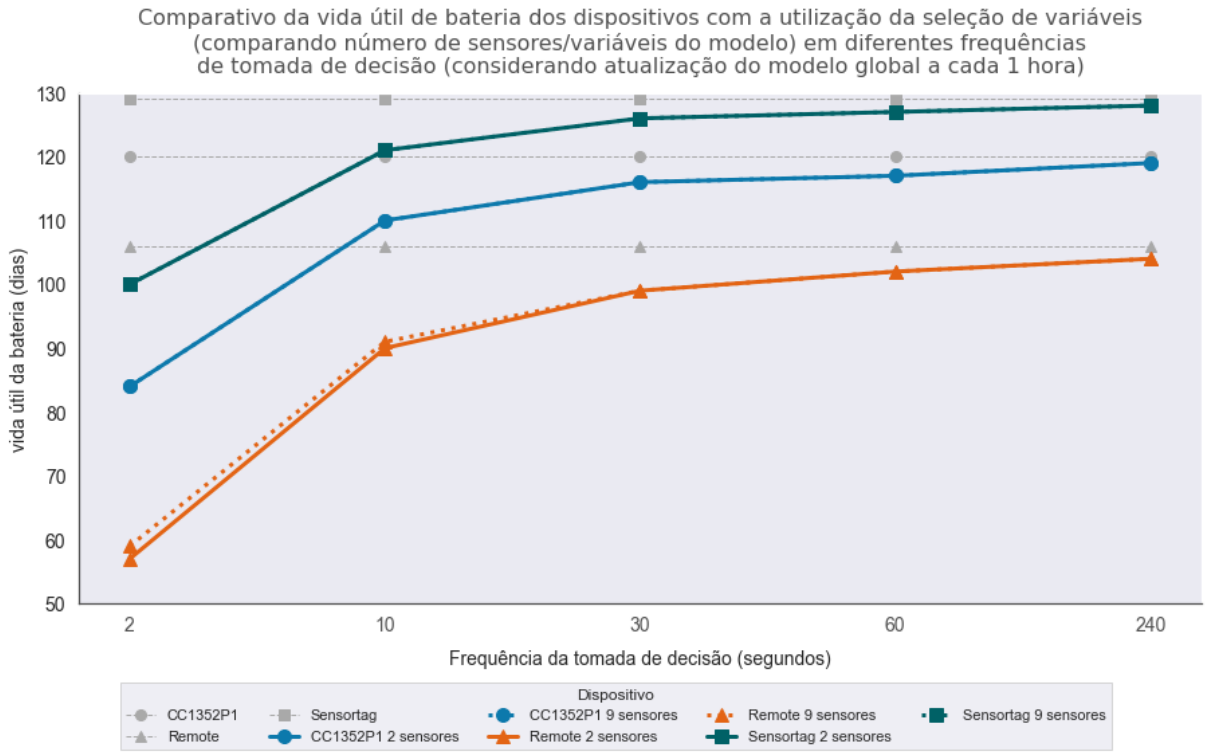
A seleção de variáveis em arquiteturas de ML em geral tem como propósito encontrar as melhores variáveis que resultam no aumento da acurácia no desfecho dos modelos.

Aplicado ao FedSensor, a seleção de variáveis tem a intenção de observar o comportamento de consumo de energia com relação ao número de variáveis utilizadas no modelo de ML global e do número de classes/grupos quando o modelo a ser utilizado é de classificação ou agrupamento. Dessa maneira pode-se comparar o consumo de energia

desses diferentes fatores.

A aplicação da seleção de variáveis (conforme apresentado no Capítulo 5), resulta na seleção das variáveis PM 2.5 e PM10, do conjunto de nove variáveis disponíveis (PM 2.5, PM 10, NO, NO_2, NO_x, NH_3, CO, SO_2, O_3).

Figura 39 – Vida útil da bateria dos dispositivos IoT ultra-restritos comparativamente com o uso de nove e duas variáveis preditoras, considerando diferentes intervalos de tempo para inferência (em segundos) e atualização do modelo global a cada uma hora



Fonte: Autor

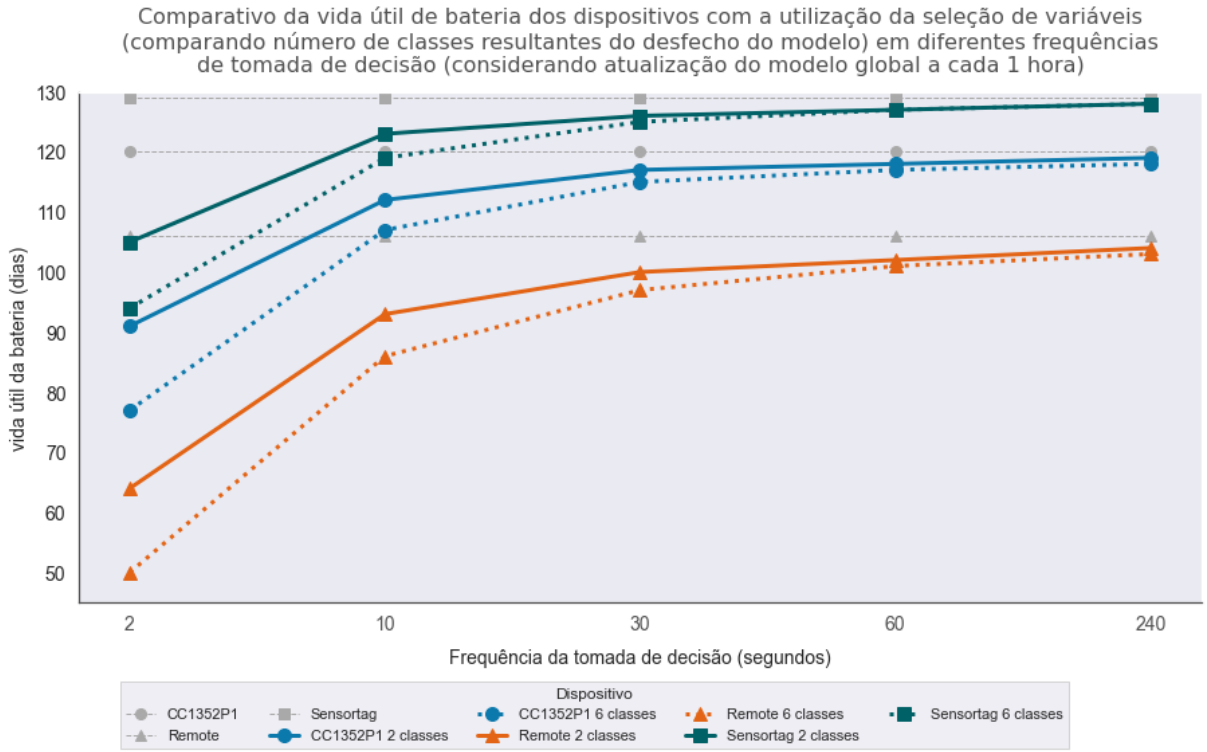
De acordo com o resultado da seleção de variáveis, a Figura 39 apresenta a vida útil das baterias dos dispositivos IoT ultra-restritos considerando um cenário sem a seleção de variáveis (com um provisionamento exigindo a coleta, transmissão de modelo de ML global e inferência considerando a utilização das nove variáveis) e com a seleção de variáveis (utilizando os sensores PM 2.5 e PM 10 resultantes da seleção de variáveis na transmissão do modelo de ML global e inferência realizada pelos dispositivos).

O comportamento do consumo de energia dos dispositivos IoT ultra-restritos se mostra similar, independentemente do número de variáveis preditoras (sensores) utilizados no modelo de ML global. O que guia o consumo de energia, conforme se observa na

Figura 39 é o tempo de inferência e não o número de sensores envolvidos para realização da previsão, uma vez que a vida útil da bateria dos dispositivos somente se altera em 1% para o dispositivo Remote com 2 e 10 segundos de realização de inferências - nos demais intervalos de tempo não há redução da vida útil.

Outro fator a ser observado é com relação ao número de classes ou grupos no desfecho do modelo de ML global (quando se usam modelos de classificação ou agrupamento). Nesse sentido, a Figura 40 mostra o comparativo da vida útil das baterias dos dispositivos com a utilização dos extremos avaliados nos cenários experimentais: 6 classes/grupos e 2 classes/grupos.

Figura 40 – Vida útil da bateria dos dispositivos IoT ultra-restritos comparativamente com o uso de seis e duas classes/grupos no desfecho, considerando diferentes intervalos de tempo para inferência (em segundos) e atualização do modelo global a cada uma hora



Fonte: Autor

Diferentemente do que se observa com o número de variáveis preditoras, o número de classes ou grupos no desfecho do modelo reduz a vida útil das baterias dos dispositivos IoT quando a inferência é de 10 ou 2 segundos, conforme se observa na Figura 40.

Observando a severa exigência de recursos nos dispositivos IoT ultra-restritos (2

segundos para realização de inferência), o uso de 6 classes ou grupos, reduz a vida útil da bateria em 35,83%, 52,83% e 27,13%, respectivamente, para os dispositivos CC1352P1, Remote e Sensortag. Para inferências realizadas a cada 30, 60 ou 240 segundos, a redução da vida útil fica entre 0,77 e 5%, dependendo do dispositivo.

Portanto, conclui-se que ao analisar as Figuras 39 e 40, o número de classes/grupos no desfecho é o fator que mais aumenta o gasto energético quando comparado ao número de sensores utilizados no modelo de ML global.

Além da análise do consumo de energia pelos dispositivos IoT ultra-restritos, avalia-se também o consumo de energia do participante (Raspberry Pi) nos experimentos conduzidos.

A Tabela 7 apresenta o consumo de energia do Raspberry Pi utilizado, obtido com a observação das diferentes tarefas durante os treinamentos federados usando o FedSensor.

Tabela 7 – Consumo de energia do participante durante as rodadas de treinamento federado

Raspberry Pi	
Ação	**Energia (mJ)**
Descanso (inativo)	236,30
Treinamento do modelo de ML local	409,84
Recebimento do modelo de ML global do gerenciador	379,77
Transmissão do modelo de ML local para o gerenciador	377,00
Transmissão do modelo de ML global para o dispositivo	376.29

Ao observar os resultados apresentados Tabela 7, o treinamento do modelo de ML é a ação que mais consome energia do dispositivo, enquanto a energia gasta no modo descanso é a ação que menos requer energia. As tarefas que envolvem a transmissão e recepção de modelos de ML apresentam consumo de energia similar, uma vez que a interface 802.11 do Raspberry Pi se mantém ligada ininterruptamente - diferente do que ocorre com os dispositivos IoT ultra-restritos, que têm como característica das redes IoT baseadas em sensores o rádio ligando e desligando em tempos sincronizados com o coordenador da rede.

Considerando os resultados de energia obtidos, a Figura 41 apresenta o consumo de energia diário, com variação do número de dispositivos nas aplicações IoT gerenciadas pelo participante (50, 100, 200 ou 300 dispositivos por aplicação IoT). Também variam-se o número de sensores utilizados nos modelos de ML (2 e 9, para demonstrar os dois extremos avaliados com relação ao número de sensores). Além disso, assim como mostrado nos resultados referentes aos dispositivos, varia-se a frequência de atualização do modelo de

ML global, nos períodos de 7,5, 15 e 30 minutos, e 1, 2, 4, 8 e 24 horas.

O consumo de energia apresentado na Figura 41 tem o objetivo de identificar o perfil energético de um participante com o aumento do volume de dados diante do número de dispositivos que compõem cada aplicação IoT. Conclui-se que o aumento médio percentual do número de dispositivos (que consequentemente aumenta a base de dados para treinamento no participante), é de 9,23% e 4,81% no consumo de energia para 7,5 e 15 minutos de atualização dos modelos de ML globais, respectivamente - em comparação com os demais períodos em que o consumo de energia é similar. Observa-se que, em um intervalo de treinamento de modelos de ML globais acima de 30 minutos, o participante fica em inatividade (de processamento e transmissão de dados) a maior parte do tempo: em média 97,92% de tempo em inatividade.

Conclui-se, ao observar os resultados apresentados referentes à seleção de variáveis, que o número de variáveis preditoras do modelo de ML global não reduz a vida útil das baterias dos dispositivos IoT ultra-restritos, nem gera um consumo excessivo de energia no participante (mesmo na presença de muitos dispositivos por aplicação IoT). O fator que reduz a vida útil das baterias dos dispositivos é o número de desfechos, uma vez que um desfecho com 2 classes/grupos garante um aumento da vida útil das baterias quando comparado a um desfecho com 6 classes/grupos, conforme se observa nas Figuras 31, 32, e 40.

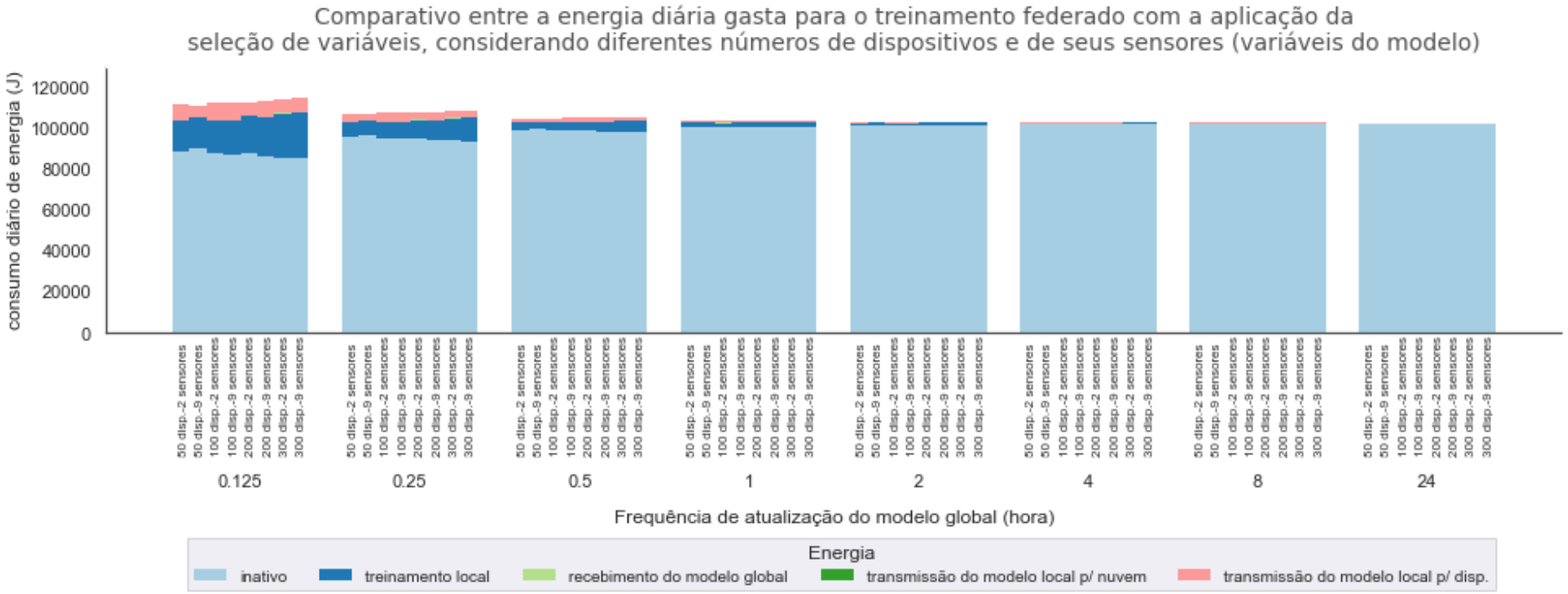
Além do observado com relação ao consumo de energia em diferentes fatores, outro recurso do FedSensor avaliado é a detecção de anomalias, cujos resultados constam na seção a seguir.

6.6 Detecção de anomalias

Com relação à detecção de anomalias, observa-se nesta seção o comportamento da função de custo do modelo de ML global regressão logística na presença de anomalias, referente ao cenário *cidades inteligentes*, no experimento *IQAr*.

As Figuras 42, 43 e 44 apresentam as curvas de aprendizado, mostrando os resultados da função de custo federado do modelo de ML regressão logística de uma aplicação IoT, respectivamente, para 2, 4 e 9 sensores. De acordo com o apresentado no Capítulo 5, as variações de anomalias apresentadas nas Figuras consideram:

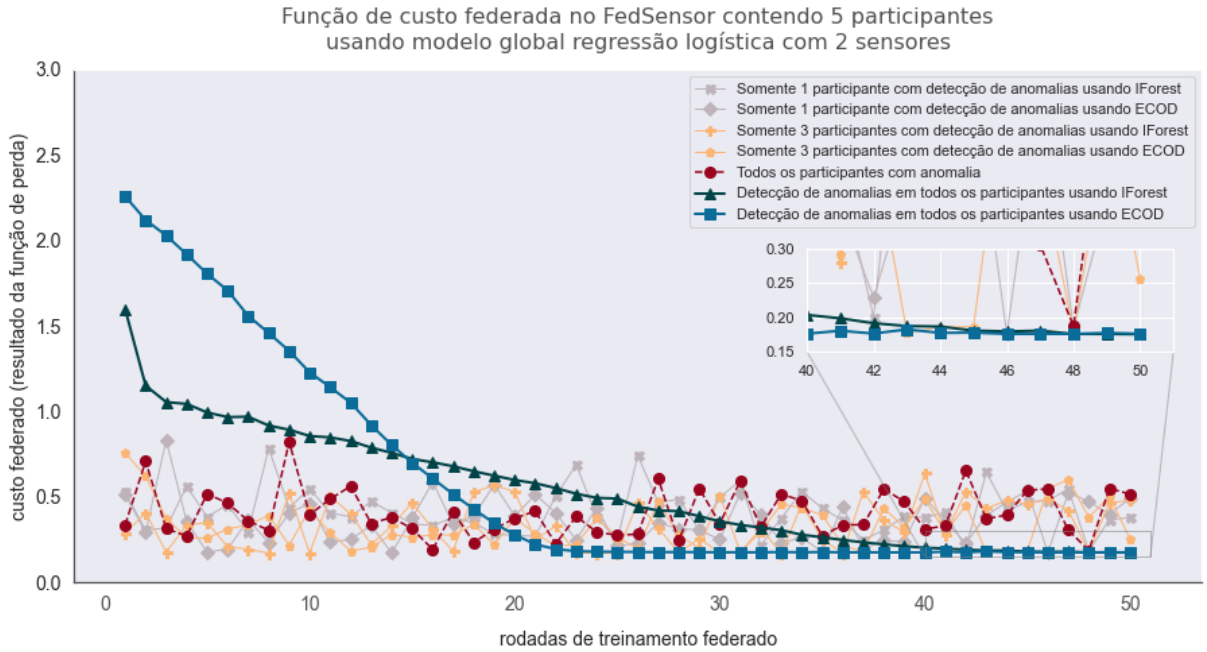
Figura 41 – Consumo de energia diário no participante do FedSensor para realização do treinamento federado considerando diferentes números de dispositivos por aplicação IoT



Fonte: Autor

- Todos os participantes com anomalias nas medições;
- Um participante com anomalias nas medições (os demais detectando anomalias aplicando iForest);
- Um participante com anomalias nas medições (os demais detectando anomalias aplicando ECOD);
- Três participantes com anomalias (os demais detectando anomalias aplicando iForest);
- Três participantes com anomalias (os demais detectando anomalias aplicando ECOD);
- Todos os participantes detectando anomalias aplicando iForest;
- Todos os participantes detectando anomalias aplicando ECOD.

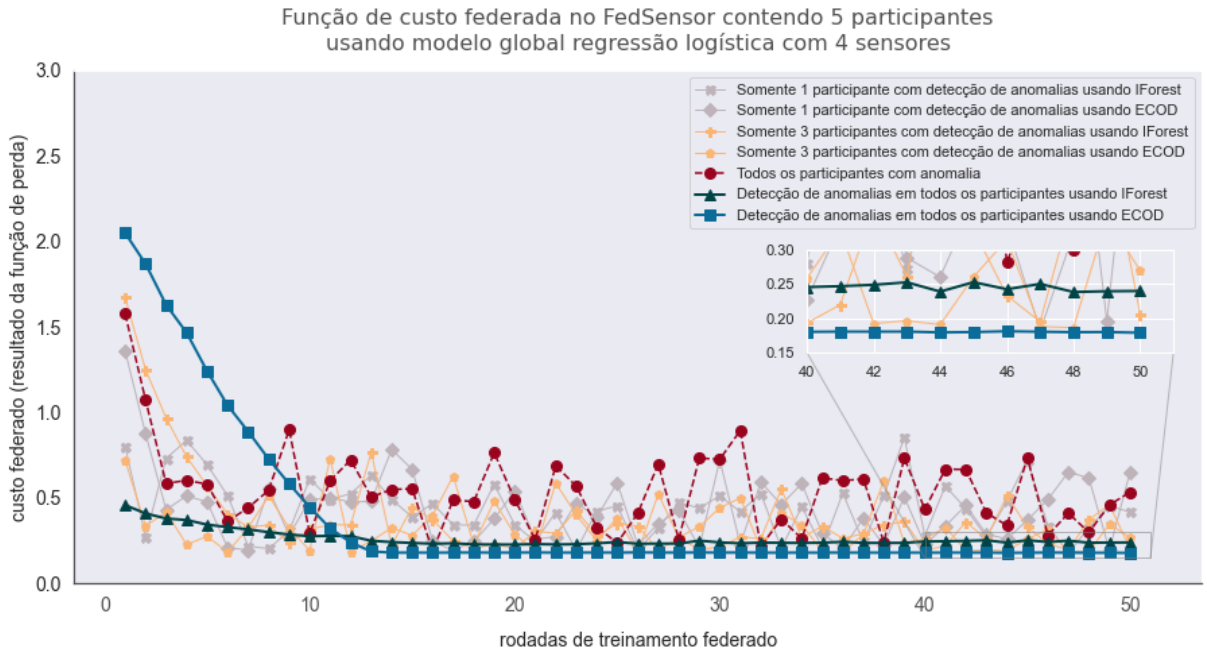
Figura 42 – Resultado da função de custo durante 50 rodadas de treinamento federado, com modelo de ML usando 2 variáveis preditoras



Fonte: Autor

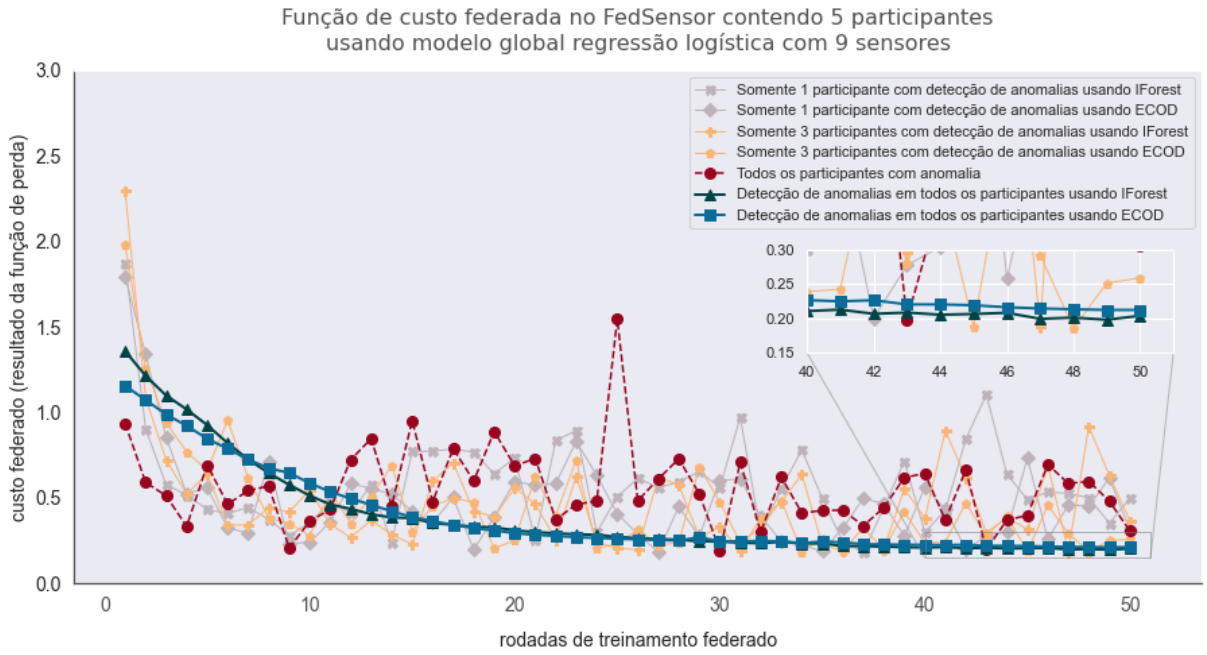
As Figuras 42, 43 e 44 evidenciam as rodadas finais do treinamento federado (40 a 50), permitindo assim comparar as diferentes condições dos experimentos relativos à detecção de anomalias.

Figura 43 – Resultado da função de custo durante 50 rodadas de treinamento federado, com modelo de ML usando 4 variáveis preditoras



Fonte: Autor

Figura 44 – Resultado da função de custo durante 50 rodadas de treinamento federado, com modelo de ML usando 9 variáveis preditoras



Fonte: Autor

Pode-se concluir, ao analisar as Figuras 42, 43 e 44 que é possível identificar medições anômalas ao observar o resultado da função de custo federado - mesmo quando apenas um participante apresenta medições anômalas. A função de custo, quando todos os participantes identificam medições anômalas, mostra um comportamento similar, pois apresenta uma redução gradual do custo conforme se executam as rodadas de treinamento federado. A presença de anomalias nas medições fica evidente pelo aumento da variabilidade no resultado da função de custo, não tendo um comportamento com redução gradual do custo quando não há presença de anomalias. Esse comportamento é similar seja com 2 (Figura 42), 4 (Figura 43) ou 9 (Figura 44) sensores.

Além da curva de aprendizado apresentada nas Figuras 42, 43 e 44, a identificação de participantes que contêm medições anômalas pode ser observada com a aplicação da Equação 4.7, cujos resultados constam nas Tabelas 8, 9 e 10.

O $IVP_{\mathcal{I}_n}$ contempla a variabilidade considerando as últimas 10 rodadas de treinamento federado, em todos os cenários apresentados nas Figuras 42, 43 e 44. Ao observar o $IVP_{\mathcal{I}_n}$ nas Tabelas 8, 9 e 10, conclui-se que a variabilidade dos participantes que contém medições anômalas está na casa de 10^{-2}, enquanto os participantes que contém medições normais, a variabilidade fica na casa de 10^{-5} a 10^{-7}, uma diferença de 10^{-3} a 10^{-5}.

Portanto, diante do apresentado, o gerenciador é capaz de identificar participantes que contém medições anômalas ao observar o $IVP_{\mathcal{I}_n}$ de cada aplicação IoT.

Tabela 8 – Resultados da variabilidade das funções de perda dos participantes do FedSensor para a identificação de anomalias - modelo de ML global com 2 variáveis (sensores)

	$IVP_{\mathcal{I}_n}$
Cenário experimental com modelo de ML global usando 2 sensores	**Resultado**
Todos os participantes com medições anômalas	0.0200567
4 participantes com medições anômalas e 1 participante detectando medições anômalas usando ECOD	0.0141088
4 participantes com medições anômalas e 1 participante detectando medições anômalas usando iForest	0.0181175
2 participantes com medições anômalas e 3 participantes detectando medições anômalas usando ECOD	0.0248273
2 participantes com medições anômalas e 3 participantes detectando medições anômalas usando iForest	0.0147732
Todos os participantes com medições normais usando ECOD na detecção	0.0000046
Todos os participantes com medições normais usando iForest na detecção	0.0000916

Tabela 9 – Resultados da variabilidade das funções de perda dos participantes do FedSensor para a identificação de anomalias - modelo de ML global com 4 variáveis (sensores)

$IVP_{\mathcal{I}_n}$	
Cenário experimental com modelo de ML global usando 4 sensores	**Resultado**
Todos os participantes com medições anômalas	0.0241467
4 participantes com medições anômalas e 1 participante detectando medições anômalas usando ECOD	0.0290370
4 participantes com medições anômalas e 1 participante detectando medições anômalas usando iForest	0.0230769
2 participantes com medições anômalas e 3 participantes detectando medições anômalas usando ECOD	0.0038125
2 participantes com medições anômalas e 3 participantes detectando medições anômalas usando iForest	0.0124284
Todos os participantes com medições normais usando ECOD na detecção	0.0000004
Todos os participantes com medições normais usando iForest na detecção	0.0000305

Tabela 10 – Resultados da variabilidade das funções de perda dos participantes do FedSensor para a identificação de anomalias - modelo de ML global com 9 variáveis (sensores)

$IVP_{\mathcal{I}_n}$	
Cenário experimental com modelo de ML global usando 9 sensores	**Resultado**
Todos os participantes com medições anômalas	0.0271800
4 participantes com medições anômalas e 1 participante detectando medições anômalas usando ECOD	0.0280575
4 participantes com medições anômalas e 1 participante detectando medições anômalas usando iForest	0.0523170
2 participantes com medições anômalas e 3 participantes detectando medições anômalas usando ECOD	0.0099046
2 participantes com medições anômalas e 3 participantes detectando medições anômalas usando iForest	0.0610764
Todos os participantes com medições normais usando ECOD na detecção	0.0000303
Todos os participantes com medições normais usando iForest na detecção	0.0000214

Os resultados apresentados permitem validar as hipóteses levantadas no Capítulo 1, com o cumprimento dos objetivos referentes ao sistema de mensagens para transmissão dos modelos de ML globais gerados no FedSensor para os dispositivos, evidenciando que a vida útil das baterias se mantém dentro da expectativa de meses de uso dos dispositivos. Condições severas de operação reduzem a vida útil, mas não inviabilizam a utilização dos dispositivos IoT ultra-restritos em arquiteturas de FL com o uso do FedSensor.

6.7 Síntese

Considerando o apresentado no Capítulo 3, observa-se a lacuna com relação à utilização de dispositivos IoT ultra-restritos em arquiteturas de FL, bem como ausência de avaliação do consumo de energia desses dispositivos com relação às frequentes atualizações dos modelos de ML globais e realizações de inferência. A Tabela 11 apresenta um comparativo entre o estado-da-arte e o FedSensor apresentado nesta Tese.

Diante do exposto, a proposta deste trabalho, que contempla a união da FL às redes IoT baseadas em sensores (com a utilização de dispositivos IoT ultra-restritos), fornece uma contribuição ao estado-da-arte ao viabilizar um novo nível de Edge Intelligence (adicionando um novo nível ao panorama apresentado por Zhou et al. (2019)). A razão

Tabela 11 – Contribuições com o estado-da-arte referente à utilização de dispositivos IoT ultra-restritos em FL

Trabalho	**Arquitetura**	**Avaliação de energia**	**Tipos de dispositivos**	**Privacidade (dados) e Anonimidade (dispositivos)**
McMahan et al. (2017)	Nuvem-dispositivos	Não	Robustos	apenas dos dados
Saha, Misra e Deb (2021)	*Fog*-dispositivos	Sim	Restritos	apenas dos dados
Ye et al. (2020)	Nuvem-*Edge*-dispositivos	Não	Restritos	apenas dos dados
Foukalas e Tziouvaras (2021)	Nuvem-*Edge*-dispositivos	Não	Restritos	apenas dos dados
Feraudo et al. (2020)	Nuvem-dispositivos	Não	Restritos	apenas dos dados
Peng et al. (2021)	Nuvem-dispositivos	Sim	Restritos	apenas dos dados
Ren et al. (2019)	*Edge*-dispositivos	Sim	Restritos	apenas dos dados
Shalaginov, Semeniuta e Alazab (2019)	Nuvem-dispositivos	Não	Ultra-restritos	sem privacidade
Este trabalho (FedSensor)	**Nuvem-*Edge*-*Extreme Edge* (dispositivos)**	**Sim**	**Ultra-restritos**	**Dados e dispositivos**

disso é que o treinamento do modelo é realizado em uma cooperação entre nuvem-edge (a camada Edge realiza o treinamento do modelo local e a camada nuvem gera o modelo global). Além disso, o dispositivo IoT ultra restrito é capaz de participar da arquitetura de FL realizando a inferência, logo depois de receber os novos modelos globais gerados. Dessa maneira, os dados dos dispositivos IoT na camada Extreme Edge não trafegam para a nuvem; eles continuam dentro do raio de alcance da própria aplicação (hospital, fazenda, entre outros) na camada Edge, mantendo assim a privacidade dos dados no ambiente em que ele está implantado. Ainda, contribuições adicionais contemplam a identificação de participantes com medições anômalas.

7 CONCLUSÃO

Neste trabalho apresenta-se o FedSensor, um *framework* de FL em redes IoT baseadas em sensores, que utiliza dispositivos IoT ultra-restritos e considera a cooperação entre nuvem e Edge para a geração de modelos de ML globais. Os objetivos levantados no Capítulo 1 consideram (1) projetar e avaliar um sistema de mensagens seguro fim-a-fim entre dispositivos IoT ultra-restritos (na camada *Extreme Edge*) e um participante (na camada *Edge*); (2) projetar e avaliar a camada *Edge Intelligence* do FedSensor para prover privacidade e anonimidade para os dispositivos IoT participantes das aplicações IoT; (3) avaliar o consumo de energia dos dispositivos IoT ultra-restritos para a recepção de diferentes modelos de ML globais, com o dispositivo em operação; (4) detectar a presença de medições anômalas durante o treinamento dos modelos de ML; e (5) avaliar o consumo de energia nas camadas *Edge* e *Extreme Edge* com a realização dos treinamentos federados e a realização de inferências pelos dispositivos IoT ultra-restritos.

As hipóteses levantadas apontam para avaliar a viabilidade do uso de dispositivos IoT ultra-restritos em arquiteturas de Federated Learning (FL), uma vez que ao se observar os trabalhos mais diretamente relacionados no Capítulo 3, os dispositivos IoT ultra-restritos não são utilizados em FL.

Dentro deste contexto, o FedSensor apresentado neste trabalho é extensamente avaliado, tanto com relação às capacidades de recepção e atualização dos modelos de ML globais pelos dispositivos IoT ultra-restritos, quanto com relação à capacidade de processamento desses dispositivos para realização de inferências, considerando diferentes condições de operação: (1) variando o número de sensores utilizados nos dispositivos (que são as variáveis preditoras dos modelos de ML), e (2) variando o número de desfechos dos modelos relativos à classificação e agrupamento.

Os resultados apresentados no Capítulo 6 permitem concluir que o FedSensor é uma alternativa viável para a utilização de dispositivos IoT ultra-restritos em arquiteturas de FL, viabilizando a tomada de decisão inteligente e autônoma por esses dispositivos, sem que seja necessário aguardar um comando da plataforma em nuvem. A espera de comandos provenientes da nuvem pode inviabilizar várias aplicações IoT, como um ajuste de postura, o acionamento de ventilação em caso de condições perigosas do ar, ou a desativação de um

motor em caso de falha. Considerando as diversas aplicações IoT existentes, observa-se a necessidade de diferentes intervalos de tempo para coleta de medições dos sensores e realização de inferências pelos dispositivos, condições avaliadas no Capítulo 6.

Ao observar os resultados apresentados no Capítulo 6 pode-se observar em quais condições do FedSensor apresenta o menor consumo de energia fornecendo aos dispositivos IoT ultra-restritos a possibilidade de participarem de uma arquitetura de FL. Pode-se concluir que o uso de menos classes ou grupos no desfecho dos modelos de ML globais corrobora para o aumento da vida útil das baterias dos dispositivos. Portanto, durante a seleção de variáveis dos modelos de ML, uma etapa para avaliação dos desfechos possíveis do modelo de ML é uma tarefa importante a ser observada para reduzir o consumo de energia nos dispositivos.

Ao comparar o consumo de energia do recebimento, estruturação e atualização dos modelos de ML globais nos dispositivos IoT ultra-restritos (observado nas Figuras 18 a 29), com o consumo de energia para a realização da inferência (observado nas Figuras 30 a 34), é possível concluir que o consumo individual de energia para a realização de inferência pelos dispositivos IoT ultra-restritos é menor. Contudo, o volume de realizações de ações de inferência faz com que o consumo de energia para as tomadas de decisão é o fator que guia a vida útil dos dispositivos, conforme se observa nas Figuras 35 a 40.

Além das questões referentes ao consumo de energia e vida útil das baterias, é comum que os dispositivos IoT ultra-restritos apresentem medições anômalas, como apresentado nos Capítulos 2, 3 e 4. Nesse sentido, o FedSensor tem a finalidade de observar o comportamento dos participantes durante o treinamento federado para identificar a origem das anomalias. Os resultados apresentados no Capítulo 6 demonstram que é possível identificar os participantes que contém medições anômalas ao se observar o IVP, que considera a função de custo federado durante a realização do treinamento. O comportamento de participantes que realizam a detecção de medições anômalas apresenta uma redução gradual da função de custo - o oposto ocorre com participantes que tenham medições anômalas, pois a função de custo apresenta resultados irregulares, com muita variabilidade em cada rodada de treinamento federado (mesmo apenas um participante contendo medições anômalas).

Diante do exposto, conclui-se que é vantajoso utilizar o FedSensor (em comparação

aos trabalhos que usam modelos de ML fixos) pois pode-se alterar o modelo de ML utilizado (seja o algoritmo ou os coeficientes, bem como os sensores utilizados) em tempo de execução, por meio das mensagens LWPubSub. Por isso no Capítulo 5 avalia-se o FedSensor em diferentes cenários de números de sensores e desfechos do modelo de ML global. Também é possível concluir que a frequência de troca do modelo global não influencia a vida útil das baterias do dispositivo, mesmo quando a troca do modelo global ocorre a cada 7,5 minutos, o que é considerado um tempo extremamente baixo e não usual em arquiteturas de FL. Um achado de pesquisa importante identificou que o que mais consome energia dos dispositivos IoT ultra-restritos é a frequência com que as decisões são tomadas (quantas vezes se executa um modelo de ML para tomada de decisão) e não a frequência com a qual o modelo de ML global é recebido pelo dispositivos IoT.

Além disso, conforme apresentado no Capítulo 3, há perda de privacidade do dispositivo IoT nos trabalhos identificados no estado-da-arte, pois o gerenciador precisa ter acesso aos dispositivos IoT. Em oposto, no FedSensor o treinamento ocorre a partir do uso dos dados existentes em cada aplicação IoT federada, sendo que esses dados são constantemente atualizados com as medições dos dispositivos que compõem aquela aplicação. Ao mesmo tempo que isso fornece a privacidade - já que os dados não deixam o ambiente inteligente em que estão implantados - também blinda os dispositivos de acesso externo ao ambiente inteligente, pois a nuvem não tem controle dos dispositivos que participam da FL, apenas a camada *Edge*.

Em resumo, os resultados permitem concluir que o FedSensor viabiliza a utilização de FL em dispositivos IoT ultra-restritos, que por sua vez recebem os modelos de ML e realizam a inferência, tomando decisões baseadas nos dados coletados por seus sensores, com a vida útil das baterias dos dispositivos viabilizando meses de operação. O FedSensor também propicia segurança pois o gerenciador não tem acesso aos dispositivos IoT ultra-restritos, os quais somente são conhecidos (e gerenciados) pelo participante, que está no mesmo ambiente inteligente que os dispositivos. Além disso, o IVP permite identificar participantes que contém medições anômalas.

Os resultados coletados permitem identificar ainda trabalhos futuros, que são a seguir apresentados.

7.1 Contribuições e trabalhos futuros

Em comparação aos trabalhos mais diretamente relacionados, identificados no Capítulo 3, o FedSensor é o primeiro *framework* a integrar dispositivos IoT ultra-restritos em arquiteturas de FL, com a atualização e modificação dos modelos de ML utilizados com o dispositivo em operação, além de prover a identificação de participantes que contém medições anômalas.

Para atingir esses objetivos, diferentes contribuições foram geradas para: o *Contiki-NG* (sistema operacional dos dispositivos IoT ultra-restritos), a *FIWARE* (plataforma de serviços *Edge*), o *flower* (software para treinamento federado), das quais apresentam-se as principais:

- Definição e padronização de uma identificação única de dispositivos IoT ultra-restritos (e de seus sensores) participantes de redes IoT baseadas em sensores baseando-se no padrão fornecido pela IPSO Alliance;
- Implementação da padronização da identificação única no Contiki-NG e na plataforma FIWARE;
- Desenvolvimento de sistema de mensagens segura fim-a-fim usando MQTT (LWPub-Sub) para tráfego de medições dos sensores dos dispositivos, usando a padronização apresentada anteriormente;
- Integração do LWPubSub ao Contiki-NG para compilação em três diferentes plataformas de hardware (CC1352P1, Remote e Sensortag);
- Extensão do LWPubSub para o tráfego de modelos de ML regressão logística, regressão linear e k-means;
- Integração entre dispositivos executando o Contiki-NG e a plataforma FIWARE usando confidencialidade, integridade e autenticidade utilizando o algoritmo de segurança AES-CCM-128;
- Implementação de IoT Agent para MQTT (baseado no IoT Agent para MQTT fornecido pela plataforma FIWARE) para transmissão e recepção de dados dos dispositivos IoT que executam o Contiki-NG;

- Implementação da fase de testes (inferência) dos modelos de ML regressão logística, regressão linear e k-means no Contiki-NG, para as três plataformas de hardware utilizadas;
- Implementação de módulo para integração entre os modelos de ML gerados no *flower* e a plataforma FIWARE;
- Implementação de módulo para recebimento dos modelos de ML gerados no Contiki-NG, enviados via plataforma FIWARE;
- Projeto de identificação única padronizada de modelos de ML a serem utilizados na plataforma FIWARE e no Contiki-NG, estendendo a padronização fornecida pela IPSO para contemplar modelos de ML;
- Desenvolvimento de índice para análise de variabilidade em modelos de ML treinados em FL com o objetivo de identificar participantes com medições anômalas;
- Desenvolvimento de ferramenta para remoção de medições anômalas de participantes utilizando o ECOD e iForest;
- Projeto e implementação de um *framework* para integrar o Contiki-NG, a plataforma FIWARE e o treinamento federado de modelos de ML.

Com as contribuições inéditas apresentadas, observam-se os seguintes trabalhos futuros.

Primeiro, pode-se estudar a implementação de Redes Neurais Artificiais (RNA) nos dispositivos IoT ultra-restritos, modificando o sistema de mensagens LWPubSub para a transmissão de mensagens ainda maiores, e a estruturação desse modelo de ML global. A avaliação de acurácia das RNA frente aos modelos de ML globais avaliados neste trabalho também é um trabalho futuro importante, pois pode definir quando uma RNA é viável e necessária para uma aplicação IoT. O consumo de energia e a vida útil das baterias dos dispositivos nesse cenário mantém-se igualmente importante ao apresentado neste trabalho, pois não basta a aplicação IoT poder fornecer melhores resultados em relação à acurácia, por exemplo, mas ser inviável no que tange à energia necessária.

Segundo, pode-se avaliar a utilização dos dispositivos IoT ultra-restritos em aplicações de visão computacional, também avaliando-se a energia necessária.

Terceiro, a avaliação detalhada da detecção de anomalias deve ser estendida para avaliar mais cenários de presença de anomalias, bem como avaliar se é possível detectar a origem das anomalias, identificando-as se são oriundas de ataques nos dispositivos ou se são falhas nos sensores. Além disso, é importante identificar quais são as medições anômalas e quais foram os dispositivos IoT ultra-restritos que publicaram essas medições no participante, para assim poder observar falsos positivos e negativos, por exemplo.

Quarto, no caso de um gerenciamento unificado de todos os dispositivos que participam do FedSensor (e que a anonimidade não é necessária), precisa-se estabelecer um mecanismo de confiança entre a nuvem e a *Edge Intelligence*.

Quinto, pode-se estudar o armazenamento temporal (*cache*) de modelos de ML globais para aplicações sazonais diárias, nas quais os dispositivos devem receber modelos de ML globais em diferentes períodos do dia e, com isso, poderem tomar a decisão mais adequada para diferentes condições em um dia. Duas possibilidades de armazenamento temporal podem ser avaliadas: (1) no dispositivo, e; (2) no participante.

Além disso, pode-se avaliar a aplicação de outros modelos federados, como árvores de decisão, redes neurais artificiais, entre outros, com o intuito de se observar acurácia e aplicabilidade nos dispositivos IoT ultra-restritos.

7.2 Limitações

Diante dos resultados observados com o FedSensor, identificam-se as seguintes limitações e os trabalhos futuros.

Dentre as limitações, observa-se que os modelos de ML globais baseados na classificação e agrupamento com mais de 2 classes ou grupos reduzem a vida útil dos dispositivos (em comparação ao consumo de energia dos dispositivos em descanso) - mas não inviabilizam a participação em arquiteturas de FL com a realização de inferências. Por isso, uma das limitações do trabalho é a recomendação de que os modelos de ML globais utilizados por dispositivos IoT ultra-restritos tenham como desfecho situações binárias para a tomada de decisões pelos dispositivos. Decisões mais granulares, com várias possibilidades de desfecho aumentam o consumo de energia e requerem estudos futuros, anteriormente apresentados.

A utilização do FedSensor implica no gerenciamento dos dispositivos ser uma tarefa única de cada participante. Portanto, uma limitação ocorre no caso da necessidade de

um gerenciamento de dispositivos em um único ponto focal (no gerenciador) - quando a anonimidade dos dispositivos não é necessária.

7.3 Publicações relacionadas e participações em projetos de pesquisa

O desenvolvimento desta Tese gerou as publicações apresentadas a seguir. As publicações como primeiro autor são diretamente originárias dos desenvolvimentos descritos neste trabalho, sendo que as demais publicações são contribuições indiretamente relacionadas, mas foram realizadas durante a realização do doutorado.

Artigos completos publicados em periódicos (1º autor):

Journal of Cloud Computing (Capes 2019 - A2):

Ferraz Junior, N., Silva, A., Guelfi, A., Kofuji, S. (2022). Performance evaluation of publish-subscribe systems in IoT using energy-efficient and context-aware secure messages. Journal of Cloud Computing, v. 11, n. 1, p. 6, dec 2022. ISSN 2192-113X.
DOI: https://doi.org/10.1186/s13677-022-00278-6

Journal of Communication and Information Systems (Capes 2016 - B1, Capes 2019 - A4):

Ferraz Junior, N., Silva, A., Guelfi, A., Azevedo, M., Kofuji, S. (2021). Lightweight and Secure Publish-Subscribe System for Cloud-Connected Ultra Low Power IoT Devices. Journal of Communication and Information Systems, 36(1), 110–113. DOI: https://doi.org/10.14209/jcis.2021.11

Trabalhos completos publicados em anais de congresso (1º autor):

Ferraz Junior, N., Silva, A. A. A., Guelfi, A. E., Kofuji, S. T. (2021). Privacy-preserving cloud-connected IoT data using context-aware and end-to-end secure messages. Procedia Computer Science, 191, 25–32. In: The 18th International Conference on Mobile Systems and Pervasive Computing (MobiSPC)
DOI: https://doi.org/10.1016/j.procs.2021.07.007

Participação em projetos de pesquisa:

Projeto Huawei/USP (dez/2020 a abr/2021): Criação de integração segura da comunicação cloud-fog-edge para uma rede de sensores sem fio.

Projeto Huawei/USP (set/2021 a nov/2021): Desenvolvimento de *framework* usando inteligência artificial em dispositivos IoT ultra-restritos, com a implantação de algoritmos de ML nos dispositivos, bem como a atualização dos algoritmos e de seus parâmetros por meio de mensagens seguras fim-a-fim.

Livros publicados:

Ferraz Junior, N. Segurança em redes sem fio e móveis. São Paulo: Editora Senac São Paulo, 2021. e-ISBN 978-65-5536-796-6.

Ferraz Junior, N. Segurança em ambientes inteligentes e Internet of things. São Paulo: Editora Senac São Paulo, 2022. e-ISBN e-ISBN 978-85-396-3511-5.

Artigos completos publicados em periódicos (co-autor):

Almeida, Felipe C.; Guelfi, Adilson E.; Silva, Anderson A. A.; **Ferraz Junior**, Norisvaldo; Schneider, Marvin O.; Gava, Vagner L.; Kofuji, Sergio T. (2022). An outlier-based analysis for behaviour and anomaly identification on IoT sensors. International Journal of Sensor Networks, volume 39, n. 2, 106-124.
https://doi.org/10.1504/IJSNET.2022.123604

Hauy Netto de Araujo, P. H., Silva, A., **Ferraz Junior, N.**, Cabrini, F., Santiago, A., Guelfi, A., Kofuji, S. (2021). Impact of Feature Selection Methods on the Classification of DDoS Attacks using XGBoost. Journal of Communication and Information Systems, 36(1), 200-214.
https://doi.org/10.14209/jcis.2021.22

Silva, M. M., Silva, A. A. A., **Ferraz Junior, N.**, Ueda, E. T., Perreira, F. D., Santos, A. S., Guelfi, A. E., Kofuji, S. T. (2021). A Proposed Blockchain-Based Voting System with User Authentication through Biometrics. Journal of Information Security and Cryptography (Enigma), 08(01), 1–11.
https://doi.org/10.17648/jisc.v8i1.78

Borrego, A., Eduardo Guelfi, A., Aparecido Alves da Silva, A., Teixeira de Azevedo, M., **Ferraz Junior, N.**, Kofuji, S. T. (2020). Modeling and validating a secure interconnection between industrial control system and corporate network using colored petri net. Colloquium Exactarum, 12(2), 45–61. https://doi.org/10.5747/ce.2020.v12.n2.e318

Silva, A. A. A., **Ferraz Junior, N.**, Guelfi, A. E., Barboza, S. H. I., Kofuji, S. T. (2019). Grouping detection and forecasting security controls using unrestricted cooperative bargains. Computer Communications, 146(July), 155–173. https://doi.org/10.1016/j.comcom.2019.07.022

REFERÊNCIAS

ABDULRAHMAN, S. et al. A survey on federated learning: The journey from centralized to distributed on-site learning and beyond. **IEEE Internet of Things Journal**, v. 8, n. 7, p. 5476–5497, 2021. ISSN 23274662.

AHVAR, E.; ORGERIE, A.-C.; LEBRE, A. Estimating Energy Consumption of Cloud, Fog, and Edge Computing Infrastructures. **IEEE Transactions on Sustainable Computing**, IEEE, v. 7, n. 2, p. 277–288, apr 2022. ISSN 2377-3782. Disponível em: https://ieeexplore.ieee.org/document/8668812/.

AL-AMRI, R. et al. A review of machine learning and deep learning techniques for anomaly detection in iot data. **Applied Sciences (Switzerland)**, v. 11, n. 12, 2021. ISSN 20763417.

AL-FUQAHA, A.; GUIZANI, M.; MOHAMMADI, M.; ALEDHARI, M.; AYYASH, M. Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. **IEEE Communications Surveys and Tutorials**, v. 17, n. 4, p. 2347–2376, 2015. ISSN 1553877X. Disponível em: https://ieeexplore.ieee.org/document/7123563/.

AL-MASRI, E. et al. Investigating Messaging Protocols for the Internet of Things (IoT). **IEEE Access**, v. 8, p. 94880–94911, 2020. ISSN 2169-3536. Disponível em: https://ieeexplore.ieee.org/document/9090208/.

AL-TURJMAN, F.; MALEKLOO, A. Smart parking in IoT-enabled cities: A survey. **Sustainable Cities and Society**, v. 49, n. December 2018, 2019. ISSN 22106707.

ARAUJO, V.; MITRA, K.; SAGUNA, S.; ÅHLUND, C. Performance evaluation of FIWARE: A cloud-based IoT platform for smart cities. **Journal of Parallel and Distributed Computing**, Elsevier Inc., v. 132, p. 250–261, 2019. ISSN 07437315. Disponível em: https://doi.org/10.1016/j.jpdc.2018.12.010.

ASGHARI, P.; RAHMANI, A. M.; JAVADI, H. H. S. Internet of Things applications: A systematic review. **Computer Networks**, Elsevier B.V., v. 148, p. 241–261, jan 2019. ISSN 13891286. Disponível em: https://doi.org/10.1016/j.comnet.2018.12.008https://linkinghub.elsevier.com/retrieve/pii/S1389128618305127.

BELGIU, M.; DRăGUţ, L. Random forest in remote sensing: A review of applications and future directions. **ISPRS Journal of Photogrammetry and Remote Sensing**, v. 114, p. 24–31, apr 2016. ISSN 09242716. Disponível em: https://linkinghub.elsevier.com/retrieve/pii/S0924271616000265.

BELLAVISTA, P. et al. A survey on fog computing for the Internet of Things. **Pervasive and Mobile Computing**, Elsevier B.V., v. 52, p. 71–99, 2019. ISSN 15741192. Disponível em: https://doi.org/10.1016/j.pmcj.2018.12.007.

BEUTEL, D. J. et al. Flower: A Friendly Federated Learning Research Framework. 2020. Disponível em: http://arxiv.org/abs/2007.14390.

CAMINHA, J.; PERKUSICH, A.; PERKUSICH, M. A Smart Trust Management Method to Detect On-Off Attacks in the Internet of Things. **Security and Communication Networks**, v. 2018, p. 1–10, 2018. ISSN 1939-0114. Disponível em: https://www.hindawi.com/journals/scn/2018/6063456/.

CHAUDHARY, A.; PEDDOJU, S. K.; KADARLA, K. Study of Internet-of-Things Messaging Protocols Used for Exchanging Data with External Sources. *In*: **Proceedings - 14th IEEE International Conference on Mobile Ad Hoc and Sensor Systems, MASS 2017**. IEEE, 2017. p. 666–671. ISBN 9781538623237. Disponível em: http://ieeexplore.ieee.org/document/8108818/.

CHEN, M. et al. A Joint Learning and Communications Framework for Federated Learning over Wireless Networks. **IEEE Transactions on Wireless Communications**, v. 20, n. 1, p. 269–283, 2021. ISSN 15582248.

CHHIKARA, P.; TEKCHANDANI, R.; KUMAR, N.; GUIZANI, M.; HASSAN, M. M. Federated Learning and Autonomous UAVs for Hazardous Zone Detection and AQI Prediction in IoT Environment. **IEEE Internet of Things Journal**, IEEE, v. 8, n. 20, p. 15456–15467, 2021. ISSN 23274662.

ČOLAKOVIĆ, A.; HADŽIALIĆ, M. **Internet of Things (IoT): A review of enabling technologies, challenges, and open research issues**. 2018. 17–39 p. Disponível em: https://linkinghub.elsevier.com/retrieve/pii/S1389128618305243.

COOK, A. A.; MISIRLI, G.; FAN, Z. Anomaly Detection for IoT Time-Series Data: A Survey. **IEEE Internet of Things Journal**, v. 7, n. 7, p. 6481–6494, 2020. ISSN 23274662.

CUI, L. et al. A survey on application of machine learning for Internet of Things. **International Journal of Machine Learning and Cybernetics**, Springer Berlin Heidelberg, v. 9, n. 8, p. 1399–1417, 2018. ISSN 1868808X. Disponível em: http://dx.doi.org/10.1007/s13042-018-0834-5.

DU, X.; ZHOU, Z.; ZHANG, Y.; RAHMAN, T. Energy-efficient sensory data gathering based on compressed sensing in IoT networks. **Journal of Cloud Computing**, Journal of Cloud Computing: Advances, Systems and Applications, v. 9, n. 1, 2020. ISSN 2192113X.

DUNKELS, A.; OSTERLIND, F.; TSIFTES, N.; HE, Z. Software-based on-line energy estimation for sensor nodes. **Proceedings of the 4th Workshop on Embedded Networked Sensors, EmNets 2007**, p. 28–32, 2007.

ELAZHARY, H. Internet of Things (IoT), mobile cloud, cloudlet, mobile IoT, IoT cloud, fog, mobile edge, and edge emerging computing paradigms: Disambiguation and research directions. **Journal of Network and Computer Applications**, Elsevier Ltd, v. 128, n. October 2018, p. 105–140, 2019. ISSN 10958592. Disponível em: https://doi.org/10.1016/j.jnca.2018.10.021.

ELSTS, A.; FAFOUTIS, X.; OIKONOMOU, G.; PIECHOCKI, R.; CRADDOCK, I. TSCH Networks for Health IoT. **ACM Transactions on Internet of Things**, v. 1, n. 2, p. 1–27, apr 2020. ISSN 2691-1914. Disponível em: https://dl.acm.org/doi/10.1145/3366617.

FERAUDO, A. et al. CoLearn. *In*: **Proceedings of the Third ACM International Workshop on Edge Systems, Analytics and Networking**. New York, NY, USA: ACM, 2020. p. 25–30. ISBN 9781450371322. Disponível em: https://dl.acm.org/doi/10.1145/3378679.3394528.

FERRAZ JUNIOR, N.; SILVA, A.; GUELFI, A.; KOFUJI, S. T. IoT6Sec: reliability model for Internet of Things security focused on anomalous measurements identification with energy analysis. **Wireless Networks**, Springer US, v. 25, n. 4, p. 1533–1556, 2019. ISSN 15728196.

FERRAZ JUNIOR, N.; SILVA, A.; GUELFI, A.; AZEVEDO, M.; KOFUJI, S. Lightweight and Secure Publish-Subscribe System for Cloud-Connected Ultra Low Power IoT Devices. **Journal of Communication and Information Systems**, v. 36, n. 1, p. 110–113, 2021. ISSN 19806604.

FERRAZ JUNIOR, N.; SILVA, A. A.; GUELFI, A. E.; KOFUJI, S. T. Privacy-preserving cloud-connected IoT data using context-aware and end-to-end secure messages. **Procedia Computer Science**, Elsevier B.V., v. 191, p. 25–32, 2021. ISSN 18770509. Disponível em: https://doi.org/10.1016/j.procs.2021.07.007https://linkinghub.elsevier.com/retrieve/pii/S1877050921013995.

FERRAZ JUNIOR, N.; SILVA, A. A.; GUELFI, A. E.; KOFUJI, S. T. Performance evaluation of publish-subscribe systems in IoT using energy-efficient and context-aware secure messages. **Journal of Cloud Computing**, v. 11, n. 1, p. 6, dec 2022. ISSN 2192-113X. Disponível em: https://journalofcloudcomputing.springeropen.com/articles/10.1186/s13677-022-00278-6.

FISCHER, M.; KUMPER, D.; TONJES, R. Towards improving the Privacy in the MQTT Protocol. *In*: **2019 Global IoT Summit (GIoTS)**. IEEE, 2019. p. 1–6. ISBN 978-1-7281-2171-0. Disponível em: https://ieeexplore.ieee.org/document/8766366/.

FOUKALAS, F.; TZIOUVARAS, A. A federated machine learning protocol for fog networks. *In*: **IEEE INFOCOM 2021 - IEEE Conference on Computer Communications Workshops, INFOCOM WKSHPS 2021**. [*S.l.: s.n.*]: IEEE, 2021. ISBN 9781665404433.

FRAGA-LAMAS, P.; LOPES, S. I.; FERNÁNDEZ-CARAMÉS, T. M. Green iot and edge AI as key technological enablers for a sustainable digital transition towards a smart circular economy: An industry 5.0 use case. **Sensors**, v. 21, n. 17, 2021. ISSN 14248220.

GARCIA-FONT, V.; GARRIGUES, C.; RIFÀ-POUS, H. A comparative study of anomaly detection techniques for smart city wireless sensor networks. **Sensors (Switzerland)**, v. 16, n. 6, 2016. ISSN 14248220.

GLAROUDIS, D.; IOSSIFIDES, A.; CHATZIMISIOS, P. Survey, comparison and research challenges of IoT application protocols for smart farming. **Computer Networks**, Elsevier B.V., v. 168, p. 107037, 2020. ISSN 13891286. Disponível em: https://doi.org/10.1016/j.comnet.2019.107037.

GÓMEZ-CARMONA, O.; CASADO-MANSILLA, D.; LÓPEZ-DE-IPINA, D.; GARCIA-ZUBIA, J. Simplicity is best: Addressing the computational cost of machine learning classifiers in constrained edge devices. **ACM International Conference Proceeding Series**, 2019.

GOPINATH, S.; GHANATHE, N.; SESHADRI, V.; SHARMA, R. Compiling KB-sized machine learning models to tiny IoT devices. *In*: **Proceedings of the 40th ACM SIGPLAN Conference on Programming Language Design and Implementation - PLDI 2019**. New York, New York, USA: ACM Press, 2019. p. 79–95. ISBN 9781450367127. Disponível em: http://dl.acm.org/citation.cfm?doid=3314221.3314597.

GUDUR, G. K.; BALAJI, B. S.; PEREPU, S. K. Resource-Constrained Federated Learning with Heterogeneous Labels and Models. **The 3rd International Workshop on Artificial Intelligence of Things (AIoT'20)**, ACM SIGKDD, San Diego, CA, p. 6, nov 2020. ISSN 23318422. Disponível em: http://arxiv.org/abs/2011.03206.

Guha Roy, D.; MAHATO, B.; DE, D.; BUYYA, R. Application-aware end-to-end delay and message loss estimation in Internet of Things (IoT) — MQTT-SN protocols. **Future Generation Computer Systems**, Elsevier B.V., v. 89, p. 300–316, dec 2018. ISSN 0167739X. Disponível em: https://doi.org/10.1016/j.future.2018.06.040https://linkinghub.elsevier.com/retrieve/pii/S0167739X17329990.

GUPTA, M.; ABDELSALAM, M.; KHORSANDROO, S.; MITTAL, S. Security and Privacy in Smart Farming: Challenges and Opportunities. **IEEE Access**, IEEE, v. 8, p. 34564–34584, 2020. ISSN 21693536.

HABIBZADEH, H.; SOYATA, T.; KANTARCI, B.; BOUKERCHE, A.; KAPTAN, C. **Sensing, communication and security planes: A new challenge for a smart city system design**. Elsevier B.V., 2018. 163–200 p. Disponível em: https://doi.org/10.1016/j.comnet.2018.08.001.

HAFEEZ, T.; XU, L.; MCARDLE, G. Edge intelligence for data handling and predictive maintenance in IIoT. **IEEE Access**, v. 9, p. 49355–49371, 2021. ISSN 21693536. Disponível em: https://ieeexplore.ieee.org/document/9387301/.

HAHM, O.; BACCELLI, E.; PETERSEN, H.; TSIFTES, N. Operating Systems for Low-End Devices in the Internet of Things: A Survey. **IEEE Internet of Things Journal**, v. 3, n. 5, p. 720–734, oct 2016. ISSN 2327-4662. Disponível em: http://ieeexplore.ieee.org/document/7347318/.

HE, C. et al. FedML: A research library and benchmark for federated machine learning. **arXiv**, 2020. ISSN 23318422.

HUANG, X. et al. FedParking: A Federated Learning Based Parking Space Estimation with Parked Vehicle Assisted Edge Computing. **IEEE Transactions on Vehicular Technology**, IEEE, v. 70, n. 9, p. 9355–9368, 2021. ISSN 19399359.

IMTEAJ, A.; THAKKER, U.; WANG, S.; LI, J.; AMINI, M. H. A Survey on Federated Learning for Resource-Constrained IoT Devices. **IEEE Internet of Things Journal**, IEEE, PP, n. FEBRUARY, p. 1–1, 2021. ISSN 2327-4662. Disponível em: https://ieeexplore.ieee.org/document/9475501/.

JALALI, F.; HINTON, K.; AYRE, R.; ALPCAN, T.; TUCKER, R. S. Fog computing may help to save energy in cloud computing. **IEEE Journal on Selected Areas in Communications**, IEEE, v. 34, n. 5, p. 1728–1739, 2016. ISSN 07338716.

JAMES, G.; WITTEN, D.; HASTIE, T.; TIBSHIRANI, R. **An Introduction to Statistical Learning**. New York, NY: Springer New York, 2013. v. 103. 1–235 p. (Springer Texts in Statistics, 4). ISSN 19381751. ISBN 978-1-4614-7137-0. Disponível em: http://link.springer.com/10.1007/978-1-4614-7138-7.

KHALED, A. E.; HELAL, A.; LINDQUIST, W.; LEE, C. IoT-DDL-Device Description Language for the 'T' in IoT. **IEEE Access**, v. 6, p. 24048–24063, 2018. ISSN 21693536. Disponível em: https://ieeexplore.ieee.org/document/8334820/.

KHALED, A. E.; HELAL, S. Interoperable communication framework for bridging RESTful and topic-based communication in IoT. **Future Generation Computer Systems**, Elsevier B.V., v. 92, p. 628–643, mar 2019. ISSN 0167739X. Disponível em: https://doi.org/10.1016/j.future.2017.12.042https://linkinghub.elsevier.com/retrieve/pii/S0167739X17317387.

KIM, G.; KANG, S.; PARK, J.; CHUNG, K. An MQTT-Based Context-Aware Autonomous System in oneM2M Architecture. **IEEE Internet of Things Journal**, IEEE, v. 6, n. 5, p. 8519–8528, 2019. ISSN 23274662.

KONEČNÝ, J.; MCMAHAN, H. B.; RAMAGE, D.; RICHTÁRIK, P. Federated Optimization: Distributed Machine Learning for On-Device Intelligence. p. 1–38, 2016. Disponível em: http://arxiv.org/abs/1610.02527.

KOVACS, E. et al. Standards-Based Worldwide Semantic Interoperability for IoT. **IEEE Communications Magazine**, v. 54, n. 11, p. 40–46, 2016. ISSN 01636804.

KUMAR, A.; GOYAL, S.; VARMA, M. Resource-efficient machine learning in 2 KB RAM for the Internet of Things. **34th International Conference on Machine Learning, ICML 2017**, v. 4, p. 3062–3071, 2017.

LEE, J. et al. Privacy-Preserving Patient Similarity Learning in a Federated Environment: Development and Analysis. **JMIR Medical Informatics**, v. 6, n. 2, p. e20, apr 2018. ISSN 2291-9694. Disponível em: http://medinform.jmir.org/2018/2/e20/.

LI, F. et al. System Statistics Learning-Based IoT Security: Feasibility and Suitability. **IEEE Internet of Things Journal**, v. 6, n. 4, p. 6396–6403, 2019. ISSN 23274662.

LI, L.; FAN, Y.; TSE, M.; LIN, K. Y. A review of applications in federated learning. **Computers and Industrial Engineering**, v. 149, n. September, 2020. ISSN 03608352.

LI, Q. et al. A Survey on Federated Learning Systems: Vision, Hype and Reality for Data Privacy and Protection. **arXiv**, p. 1–46, jul 2021. ISSN 23318422. Disponível em: http://arxiv.org/abs/1907.09693.

LI, Z. et al. ECOD: Unsupervised Outlier Detection Using Empirical Cumulative Distribution Functions. **IEEE Transactions on Knowledge and Data Engineering**, IEEE, v. 14, n. 8, 2022. ISSN 15582191.

LIM, W. Y. B. et al. Federated Learning in Mobile Edge Networks: A Comprehensive Survey. **IEEE Communications Surveys and Tutorials**, IEEE, v. 22, n. 3, p. 2031–2063, 2020. ISSN 1553-877X. Disponível em: https://ieeexplore.ieee.org/document/9060868/.

LIN, J. et al. MCUNet: Tiny Deep Learning on IoT Devices. p. 1–13, jul 2020. Disponível em: http://arxiv.org/abs/2007.10319.

LIU, K. et al. Automated Feature Selection: A Reinforcement Learning Perspective. **IEEE Transactions on Knowledge and Data Engineering**, v. 4347, n. c, 2021. ISSN 15582191.

LU, X.; LIAO, Y.; LIO, P.; HUI, P. Privacy-preserving asynchronous federated learning mechanism for edge network computing. **IEEE Access**, IEEE, v. 8, p. 48970–48981, 2020. ISSN 21693536.

MADDIKUNTA, P. K. R. et al. Industry 5.0: A survey on enabling technologies and potential applications. **Journal of Industrial Information Integration**, Elsevier Inc., v. 26, n. February 2021, 2022. ISSN 2452414X.

MARGI, C. B.; ALVES, R. C. A.; SEPULVEDA, J. Sensing as a Service: Secure Wireless Sensor Network Infrastructure Sharing for the Internet of Things. **Open Journal of Internet of Things**, v. 3, n. 1, 2017. ISSN 2364-7108. Disponível em: http://www.ronpub.com/ojiot.

MARTINEZ, B.; MONTÓN, M.; VILAJOSANA, I.; PRADES, J. D. The Power of Models: Modeling Power Consumption for IoT Devices. **IEEE Sensors Journal**, IEEE, v. 15, n. 10, p. 5777–5789, 2015. ISSN 1530437X.

MCMAHAN, B.; MOORE, E.; RAMAGE, D.; HAMPSON, S.; AGUERA Y ARCAS, B. Communication-Efficient Learning of Deep Networks from Decentralized Data. *In*: SINGH, A.; ZHU, J. (ed.). **Proceedings of the 20th International Conference on Artificial Intelligence and Statistics**. [*S.l.: s.n.*]: PMLR, 2017. (Proceedings of Machine Learning Research, v. 54), p. 1273–1282.

MERENDA, M.; PORCARO, C.; IERO, D. Edge machine learning for ai-enabled iot devices: A review. **Sensors (Switzerland)**, v. 20, n. 9, p. 1–34, 2020. ISSN 14248220.

MIORANDI, D.; SICARI, S.; De Pellegrini, F.; CHLAMTAC, I. Internet of things: Vision, applications and research challenges. **Ad Hoc Networks**, Elsevier B.V., v. 10, n. 7, p. 1497–1516, sep 2012. ISSN 15708705. Disponível em: http://dx.doi.org/10.1016/j.adhoc.2012.02.016http://linkinghub.elsevier.com/retrieve/pii/S1570870512000674.

MONTORI, F.; BEDOGNI, L.; Di Felice, M.; BONONI, L. **Machine-to-machine wireless communication technologies for the Internet of Things: Taxonomy, comparison and open issues**. Elsevier B.V., 2018. 56–81 p. Disponível em: https://doi.org/10.1016/j.pmcj.2018.08.002.

MORIN, É.; MAMAN, M.; GUIZZETTI, R.; DUDA, A. Comparison of the Device Lifetime in Wireless Networks for the Internet of Things. **IEEE Access**, v. 5, p. 7097–7117, 2017. ISSN 21693536.

NAKAGAWA, I.; SHIMOJO, S. IoT Agent Platform Mechanism with Transparent Cloud Computing Framework for Improving IoT Security. *In*: **2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC)**. IEEE, 2017. p. 684–689. ISBN 978-1-5386-0367-3. Disponível em: http://ieeexplore.ieee.org/document/8030012/.

NASSER, N.; KARIM, L.; ALI, A.; ANAN, M.; KHELIFI, N. Routing in the internet of things. **2017 IEEE Global Communications Conference, GLOBECOM 2017 - Proceedings**, v. 2018-Janua, p. 1–6, 2017.

NGUYEN, T. D. et al. DÏoT: A federated self-learning anomaly detection system for IoT. **Proceedings - International Conference on Distributed Computing Systems**, IEEE, v. 2019-July, p. 756–767, 2019.

NI, J.; LIN, X.; SHEN, X. S. Toward Edge-Assisted Internet of Things: From Security and Efficiency Perspectives. **IEEE Network**, IEEE, v. 33, n. 2, p. 50–57, 2019. ISSN 1558156X.

NIKNAM, S.; DHILLON, H. S.; REED, J. H. Federated Learning for Wireless Communications: Motivation, Opportunities, and Challenges. **IEEE Communications Magazine**, v. 58, n. 6, p. 46–51, jun 2020. ISSN 0163-6804. Disponível em: https://ieeexplore.ieee.org/document/9141214/.

NISHIO, T.; YONETANI, R. Client Selection for Federated Learning with Heterogeneous Resources in Mobile Edge. *In*: **ICC 2019 - 2019 IEEE International Conference on Communications (ICC)**. IEEE, 2019. p. 1–7. ISBN 978-1-5386-8088-9. ISSN 23318422. Disponível em: https://ieeexplore.ieee.org/document/8761315/.

OIKONOMOU, G. et al. The Contiki-NG open source operating system for next generation IoT devices. **SoftwareX**, Elsevier B.V., v. 18, p. 101089, 2022. ISSN 23527110. Disponível em: https://doi.org/10.1016/j.softx.2022.101089.

OMA LightweightM2M (LWM2M). **OMA (Open Mobile Alliance) Specification**. 2017. Disponível em: http://technical.openmobilealliance.org/Technical/technical-information/release-program/current-releases/oma-lightweightm2m-v1-0-2.

PENG, C. et al. Energy-Efficient Device Selection in Federated Edge Learning. *In*: **2021 International Conference on Computer Communications and Networks (ICCCN)**. IEEE, 2021. v. 2021-July, p. 1–9. ISBN 978-1-6654-1278-0. ISSN 10952055. Disponível em: https://ieeexplore.ieee.org/document/9522303/.

PORTILLA, J.; MUJICA, G.; LEE, J.-S.; RIESGO, T. The Extreme Edge at the Bottom of the Internet of Things: A Review. **IEEE Sensors Journal**, IEEE, v. 19, n. 9, p. 3179–3190, may 2019. ISSN 1530-437X. Disponível em: https://ieeexplore.ieee.org/document/8607067/.

POSNER, J.; TSENG, L.; ALOQAILY, M.; JARARWEH, Y. Federated Learning in Vehicular Networks: Opportunities and Solutions. **IEEE Network**, v. 35, n. 2, p. 152–159, 2021. ISSN 1558156X.

QI, J.; ZHOU, Q.; LEI, L.; ZHENG, K. Federated reinforcement learning: techniques, applications, and open challenges . **Intelligence & Robotics**, p. 1–39, 2021.

QIN, Z.; Ye Li, G.; YE, H. Federated Learning and Wireless Communications. **IEEE Wireless Communications**, IEEE, PP, p. 1–7, 2021. ISSN 15580687.

RAZA, S.; HELGASON, T.; PAPADIMITRATOS, P.; VOIGT, T. SecureSense: End-to-end secure communication architecture for the cloud-connected Internet of Things. **Future Generation Computer Systems**, Elsevier B.V., v. 77, p. 40–51, 2017. ISSN 0167739X. Disponível em: http://dx.doi.org/10.1016/j.future.2017.06.008.

RAZA, S.; WALLGREN, L.; VOIGT, T. SVELTE: Real-time intrusion detection in the Internet of Things. **Ad Hoc Networks**, v. 11, n. 8, p. 2661–2674, nov 2013. ISSN 15708705. Disponível em: http://linkinghub.elsevier.com/retrieve/pii/S1570870513001005.

REN, H.; ANICIC, D.; RUNKLER, T. TinyOL: TinyML with Online-Learning on Microcontrollers. *In*: **International Joint Conference on Neural Network (IJCNN)**. [*S.l.: s.n.*], 2021. Disponível em: http://arxiv.org/abs/2103.08295.

REN, J.; WANG, H.; HOU, T.; ZHENG, S.; TANG, C. Federated Learning-Based Computation Offloading Optimization in Edge Computing-Supported Internet of Things. **IEEE Access**, IEEE, v. 7, p. 69194–69201, may 2019. ISSN 2169-3536. Disponível em: https://ieeexplore.ieee.org/document/8761315/https://ieeexplore.ieee.org/document/8728285/.

RIEKSTIN, A. C. et al. A Survey on Metrics and Measurement Tools for Sustainable Distributed Cloud Networks. **IEEE Communications Surveys and Tutorials**, v. 20, n. 2, p. 1244–1270, 2018. ISSN 1553-877X. Disponível em: https://ieeexplore.ieee.org/document/8226747/.

RUBIN, F. P. et al. Evaluating Energy and Thermal Efficiency of Anomaly Detection Algorithms in Edge Devices. **International Conference on Information Networking**, IEEE, v. 2020-January, p. 208–213, 2020. ISSN 19767684.

RUSSELL, S. J.; NORVIG, P.; EDWARDS, D. D.; HAY, N. J.; MALIK, J. M. **Artificial Intelligence: A Modern Approach**. [*S.l.: s.n.*]: Pearson Education, 2016. 1145 p. ISBN 9781292153964.

SAHA, R.; MISRA, S.; DEB, P. K. FogFL: Fog-Assisted Federated Learning for Resource-Constrained IoT Devices. **IEEE Internet of Things Journal**, v. 8, n. 10, p. 8456–8463, 2021. ISSN 23274662.

SAMAILA, M. G.; NETO, M.; FERNANDES, D. A. B.; FREIRE, M. M.; INÁCIO, P. R. M. Challenges of securing Internet of Things devices: A survey. **Security and Privacy**, v. 1, n. 2, p. e20, mar 2018. ISSN 24756725. Disponível em: http://doi.wiley.com/10.1002/spy2.20.

SAMPAIO, G. S.; FILHO, A. R. d. A. V.; SILVA, L. S. da; SILVA, L. A. da. Prediction of motor failure time using an artificial neural network. **Sensors (Switzerland)**, v. 19, n. 19, p. 5–7, 2019. ISSN 14248220.

SARKAR, D.; NARANG, A.; RAI, S. Fed-focal loss for imbalanced data classification in federated learning. **arXiv**, 2020. ISSN 23318422.

SATTLER, F.; WIEDEMANN, S.; MULLER, K. R.; SAMEK, W. Robust and Communication-Efficient Federated Learning from Non-i.i.d. Data. **IEEE Transactions on Neural Networks and Learning Systems**, v. 31, n. 9, p. 3400–3413, 2020. ISSN 21622388.

SHA, K.; WEI, W.; Andrew Yang, T.; WANG, Z.; SHI, W. On security challenges and open issues in Internet of Things. **Future Generation Computer Systems**, Elsevier B.V., v. 83, p. 326–337, 2018. ISSN 0167739X. Disponível em: https://doi.org/10.1016/j.future.2018.01.059.

SHALAGINOV, A.; SEMENIUTA, O.; ALAZAB, M. MEML: Resource-aware MQTT-based Machine Learning for Network Attacks Detection on IoT Edge Devices. **UCC 2019 Companion - Proceedings of the 12th IEEE/ACM International Conference on Utility and Cloud Computing**, p. 123–128, 2019.

SHELBY, Z.; BORMANN, C. **6LoWPAN: the wireless embedded internet**. [*S.l.: s.n.*]: Wiley, 2011. 217 p. ISBN 9780470747995.

SILVA, B. N.; KHAN, M.; HAN, K. Towards sustainable smart cities: A review of trends, architectures, components, and open challenges in smart cities. **Sustainable Cities and Society**, Elsevier, v. 38, n. February, p. 697–713, 2018. ISSN 22106707. Disponível em: https://doi.org/10.1016/j.scs.2018.01.053.

SLIWA, B.; PIATKOWSKI, N.; WIETFELD, C. LIMITS: Lightweight Machine Learning for IoT Systems with Resource Limitations. *In*: **ICC 2020 - 2020 IEEE International Conference on Communications (ICC)**. IEEE, 2020. v. 2020-June, n. Icc, p. 1–7. ISBN 978-1-7281-5089-5. ISSN 15503607. Disponível em: https://ieeexplore.ieee.org/document/9149180/.

SUGUMARAN, V.; MURALIDHARAN, V.; RAMACHANDRAN, K. I. Feature selection using Decision Tree and classification through Proximal Support Vector Machine for fault diagnostics of roller bearing. **Mechanical Systems and Signal Processing**, v. 21, n. 2, p. 930–942, 2007. ISSN 08883270.

SUTTON, R.; BARTO, A. **Reinforcement Learning: An Introduction**. Second. The MIT Press, 1998. v. 9. 1054–1054 p. ISSN 1045-9227. ISBN 0262039249. Disponível em: http://ieeexplore.ieee.org/document/712192/.

USLU, B. Ç.; OKAY, E.; DURSUN, E. Analysis of factors affecting IoT-based smart hospital design. **Journal of Cloud Computing**, v. 9, n. 1, 2020. ISSN 2192113X.

VILAJOSANA, X. et al. A realistic energy consumption model for TSCH networks. **IEEE Sensors Journal**, IEEE, v. 14, n. 2, p. 482–489, 2014. ISSN 1530437X.

WANG, H.; LI, J.; HE, K. Hierarchical ensemble reduction and learning for resource-constrained computing. **ACM Transactions on Design Automation of Electronic Systems**, v. 25, n. 1, 2019. ISSN 15577309.

WANG, S. et al. Adaptive Federated Learning in Resource Constrained Edge Computing Systems. **IEEE Journal on Selected Areas in Communications**, IEEE, v. 37, n. 6, p. 1205–1221, jun 2019. ISSN 0733-8716. Disponível em: https://ieeexplore.ieee.org/document/8664630/.

WANG, X.; WANG, C.; LI, X.; LEUNG, V. C. M.; TALEB, T. Federated Deep Reinforcement Learning for Internet of Things With Decentralized Cooperative Edge Caching. **IEEE Internet of Things Journal**, IEEE, v. 7, n. 10, p. 9441–9455, oct 2020. ISSN 2327-4662. Disponível em: https://ieeexplore.ieee.org/document/9062302/.

WARDEN, P.; SITUNAYAKE, D. **TinyML: Machine Learning with TensorFlow Lite on Arduino and Ultra-Low-Power Microcontrollers**. 1. ed. Sebastopol: O'Reilly, 2020. 504 p. ISBN 9781492052036. Disponível em: http://oreilly.com/catalog/errata.csp?isbn=9781492052043for.

XIA, Q.; YE, W.; TAO, Z.; WU, J.; LI, Q. A survey of federated learning for edge computing: Research problems and solutions. **High-Confidence Computing**, Elsevier B.V., v. 1, n. 1, p. 100008, 2021. ISSN 26672952. Disponível em: https://doi.org/10.1016/j.hcc.2021.100008.

XU, X.; LU, Y.; VOGEL-HEUSER, B.; WANG, L. Industry 4.0 and Industry 5.0—Inception, conception and perception. **Journal of Manufacturing Systems**, Elsevier Ltd, v. 61, n. September, p. 530–535, 2021. ISSN 02786125.

YANG, W. et al. Privacy is not Free: Energy-Aware Federated Learning for Mobile and Edge Intelligence. **12th International Conference on Wireless Communications and Signal Processing, WCSP 2020**, p. 233–238, 2020.

YAO, L. et al. Compressive Representation for Device-Free Activity Recognition with Passive RFID Signal Strength. **IEEE Transactions on Mobile Computing**, IEEE, v. 17, n. 2, p. 293–306, 2018. ISSN 15361233.

YAZICI, M.; BASURRA, S.; GABER, M. Edge Machine Learning: Enabling Smart Internet of Things Applications. **Big Data and Cognitive Computing**, v. 2, n. 3, p. 26, 2018. ISSN 2504-2289.

YE, Y.; LI, S.; LIU, F.; TANG, Y.; HU, W. EdgeFed: Optimized Federated Learning Based on Edge Computing. **IEEE Access**, v. 8, p. 209191–209198, 2020. ISSN 21693536.

YIGITCANLAR, T. et al. Can cities become smart without being sustainable? A systematic review of the literature. **Sustainable Cities and Society**, Elsevier, v. 45, n. October 2018, p. 348–365, feb 2019. ISSN 22106707. Disponível em: https://doi.org/10.1016/j.scs.2018.11.033https://linkinghub.elsevier.com/retrieve/pii/S221067071831268X.

ZHOU, C. et al. Privacy-Preserving Federated Learning in Fog Computing. **IEEE Internet of Things Journal**, v. 7, n. 11, p. 10782–10793, 2020. ISSN 23274662.

ZHOU, Z. et al. Edge Intelligence: Paving the Last Mile of Artificial Intelligence With Edge Computing. **Proceedings of the IEEE**, v. 107, n. 8, p. 1738–1762, aug 2019. ISSN 0018-9219. Disponível em: https://ieeexplore.ieee.org/document/8736011/.

ZIKRIA, Y. B.; AFZAL, M. K.; ISHMANOV, F.; KIM, S. W.; YU, H. A survey on routing protocols supported by the Contiki Internet of things operating system. **Future Generation Computer Systems**, Elsevier B.V., v. 82, p. 200–219, 2018. ISSN 0167739X. Disponível em: https://doi.org/10.1016/j.future.2017.12.045.

APÊNDICES

APÊNDICE A – DISPONIBILIZAÇÃO DO CÓDIGO-FONTE

O framework FedSensor está disponível publicamente em https://github.com/norisjunior/FedSensor.

Os principais códigos desenvolvidos para o framework foram:

- Plataforma Edge (diretório EdgeServer):
  - Estrutura dos microsserviços que compõem a plataforma (arquivo *docker-compose.yml*;
  - Microsserviço para tráfego fim-a-fim entre um dispositivo IoT ultra-restrito e a plataforma Edge, por meio do IoT Agent desenvolvido (contêiner norisjunior/lwpubsub-iotagent-ccm:v2.10, disponível no Docker Hub);
  - Script para provisionamento dos dispositivos IoT ultra-restritos utilizados (CC1352P1, Remote e Sensortag) na plataforma Edge, no arquivo: *provision_-lwaiot_devices.sh*;
- Treinamento federado (diretório FL/fedsensor_framework):
  - Gerenciador, contendo o número de rodadas de treinamento federado, as portas de comunicação com os participantes, número mínimo de participantes por rodada, modelos de ML suportados, entre outras informações (arquivo manager.py);
  - Participante, contendo as aplicações IoT disponíveis e seus modelos de ML suportados (com diferentes combinações de variáveis preditoras/sensores e desfechos). Ao final do treinamento federado, o participante envia o modelo de ML global para o dispositivo IoT ultra-restritos, considerando o provisionamento realizado anteriormente na plataforma Edge (arquivo participant.py para terminal Linux e participant_pi3b.py para uso no Raspberry Pi);
  - Serialização das mensagens, considerando a utilização da padronização da IPSO Alliance (arquivos utils.py e lwpubsub_serialization.py);
  - Seleção de variáveis: avaliação dos conjuntos de dados para escolha das variáveis que mais contribuem para o desfecho (arquivo edge_feature_selection.py);

- Dispositivos IoT ultra-restritos (arquivo contiki-4.7/lwiotms/lwpubsub/lwpubsub-fedsensor-ml_and_msg.c):
  - Código utilizado em todos os dispositivos IoT ultra-restritos;
  - Contém a inferência dos modelos de ML regressão linear, regressão logística e k-means;
  - Transmissão fim-a-fim de dados para o IoT Agent da plataforma Edge (norisjunior/lwpubsub-iotagent-ccm:v2.10);
  - Recepção de modelos de ML do IoT Agent da plataforma Edge;
- Roteiro dos experimentos (arquivo experiments/Roteiro_experimentos.md):
  - Orientações gerais para utilização do framework FedSensor, considerando os cenários "Indústria 5.0" (experimento *motor*) e "Cidades inteligentes" (experimento *IQAr*).