Relatório Técnico de Laboratório de Redes: Predição de QoS de um Serviço de Streaming utilizando Aprendizado Federado

Eduardo Montagner de Moraes Sarmento e Lucas Miguel Tassis

Fevereiro de 2023

1 Introdução

Qualidade de serviço (QoS, do inglês: Quality of Service) é a medida da qualidade geral de um serviço, medindo a performance vista pelos usuários. Baixa latência, baixa variação de latência, velocidade de rede alta e baixa perda de pacotes, por exemplo, são fatores que influenciam em um serviço ter um QoS alto. Por outro lado, fatores como alta latência e alta perda de pacotes fazem com que o QoS seja baixo. Por ser um serviço real-time, streaming de vídeos tem como um requerimento fundamental que o QoS seja alto para que os usuários possam usufruir deste tipo de serviço de forma confortável. Podemos medir a QoS de um serviço de streaming mensurando algumas características do streaming durante o tempo, como, por exemplo, a perda de frames, o bitrate da transmissão, quantos segundos do video estão bufferizados, entre outros.

Podemos usar da previsão destas características para prever a QoS experimentada pelo usuário, o que é interessante para podermos saber previamente se haverá uma queda de qualidade de serviço e tenhamos a oportunidade de corrigir os problemas antes que essa queda aconteça. Para realizar esta previsão podemos utilizar algoritmos de aprendizado de máquina treinados com características da rede que liga os usuários aos servidores, como, por exemplo, a quantidade de bytes enviadas e recebidas por cada roteador no caminho entre os clientes e o servidor, e características do sistema do servidor e dos roteadores, como uso de CPU e memória RAM, por exemplo.

Porém essas características podem revelar o uso de internet do usuário e assim infringir sobre sua privacidade. Sendo assim, para realizar a previsão precisamos treinar nossos algoritmos visando manter esta privacidade. Por isso, neste trabalho treinamos os algoritmos com a técnica aprendizado federado, uma técnica de treinamento distribuída onde não há transferência dos dados na rede, mas a transferência dos pesos dos algoritmos treinados localmente, garantindo, assim, a privacidade do usuário.

Neste trabalho testamos, usando diferentes configurações de rede, o desempenho de algoritmos de aprendizado de máquina para a predição do QoS de um serviço de streaming. Para cada configuração da rede medimos segundo a segundo o uso de CPU, uso de RAM, quantidade de bytes e pacotes recebidos e enviados de cada dispositivo de rede entre os usuários e o servidor. Dessa forma, construindo nosso conjunto de dados para aquela configuração de rede. Por fim, treinamos os algoritmos de aprendizado de máquina utilizando esses conjuntos de dados localmente, e também utilizando a técnica de aprendizado federado. Com isso, comparamos os resultados obtidos por essas duas técnidas, comparando seu desempenho na predição da QoS.

O restante do documento é organizado da seguinte forma: Na Seção 2 é apresentado um breve resumo sobre aprendizado federado e também a topologia de rede utilizada. Na Seção 3 são apresentados os resultados obtidos nos experimentos realizados. E, por fim, na Seção 4 o trabalho e concluído. Também são apresentados a proposta de alguns trabalhos futuros.

2 Metodologia

Nessa seção apresentaremos um breve resumo sobre aprendizado federado. Também será introduzido a topologia de rede utilizada, bem como o *setup* experimental.

2.1 Aprendizado Federado

Aprendizado federado é uma técnica de treinamento de algoritmos de aprendizado de máquina, principalmente utilizada para o treinamento de redes neurais [3]. Esta técnica foi desenvolvida com o intuito

de permitir o treinamento eficiente de algoritmos de aprendizado de máquina em ambientes onde os dados estão distribuídos entre vários participantes do treinamento, mas onde a privacidade dos dados dos participantes é fundamental.

Podemos observar o funcionamento desta técnica, para uma arquitetura centralizada, mesma utilizada no framework flwr disponível na biblioteca flower que usamos neste trabalho, na figura 1. Nela temos um servidor e n clientes, sendo os participantes do aprendizado.

Os clientes começam inicializando seus modelos locais e se conectam ao servidor, o servidor então seleciona um número aleatório de clientes para participar do aprendizado naquele round. Durante o round os modelos locais são treinados por cada cliente com seus dados locais. Assim que o treinamento local de um cliente acaba, este cliente envia os pesos de seu modelo ao servidor. Quando todos os clientes acabam seus treinamentos, o servidor realiza a agregação dos pesos executando algum algoritmo de agregação, como, por exemplo, o federated average [4], tendo assim o modelo global para aquele round. Os pesos agregados do modelo global são enviados pelo servidor aos clientes, onde cada cliente testa o modelo global com seus dados locais. Após este processo o servidor escolhe novos clientes aleatórios, iniciando um novo round. Este processo continua até que o modelo global convirja ou um número pré-estabelecido de rounds tenha passado.

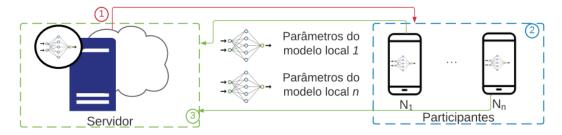


Figure 1: Arquitetura do aprendizado federado. Fonte: [3]

2.2 Topologia da rede

Nesse trabalho, simulamos uma rede cabeada usando o ambiente de emulação de redes Mininet-Wifi [2]. A rede é formada por três hosts: h1, h2 e h3; quatro switches: sw0, sw1, sw2 e sw3; e um servidor: server. A Figura 2 ilustra a topologia de rede utilizada. Cada um dos hosts está conectado com seu switch correspondente, ou seja, h1 está conectado com sw1, h2 com sw2 e h3 com sw3. Estes switches estão conectados a um switch, sw0, que faz a ponte entre os hosts e o servidor. O servidor está conectado a rede da virtual machine do Mininet-Wifi, onde um servidor Apache2 está em execução. Esse servidor é responsável por disponibilizar o arquivo do filme "Big Buck Bunny" [1] utilizado para os testes de streaming. Os hosts utilizam um player de vídeo MPEG-DASH para streaming do vídeo. A implementação desse player pode ser encontrado no seguinte link: GitHub.

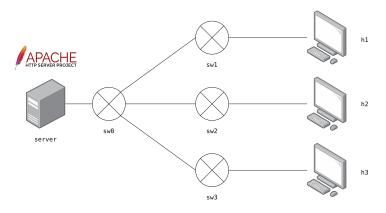


Figure 2: Ilustração da topologia da rede usada nos experimentos.

3 Resultados

Nessa seção serão apresentados os resultados obtidos em quatro experimentos gerados a partir da topologia de rede apresentada. Para cada experimento, a bandwidth e loss dos links de conexão foram modificados, a fim de criar diferentes cenários para predição da QoS.

3.1 Setup Experimental

Para cada experimento os dados de dois hosts foram utilizados para treino dos modelos e os dados do terceiro host, para teste.

As features utilizadas para treino foram obtidas a partir de um script escrito em Python, que era executado em cada switch. Esse script coletava os seguintes dados: uso de CPU (CPU_use); uso de RAM (RAM_filled); bytes enviados pela rede (Net_bytes_sent); bytes recebidos pela rede (Net_bytes_received); uso de disco (Disk_IO_percentage); número de pacotes de rede enviado (Packets_sent); e número de pacotes de rede recebido (Packets_received). Para avaliação do QoS, foram capturamos o bitrate e buffer level do player durante a stream.

Foram utilizados três modelos de aprendizado de máquina: (1) aprendizado federado (implementação da biblioteca flower); (2) rede neural artificial (implementação da biblioteca scikit-learn); e (3) floresta aleatória (implementação da biblioteca scikit-learn.

Todos os códigos desenvolvidos para esse trabalho podem ser encontrados em GitHub.

3.2 Experimento I

A Figura 3 ilustra o cenário da topologia de rede.

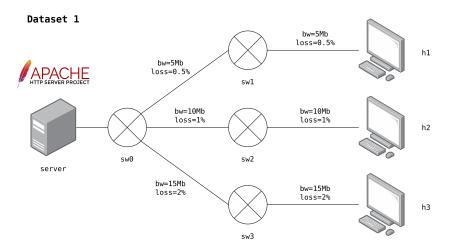
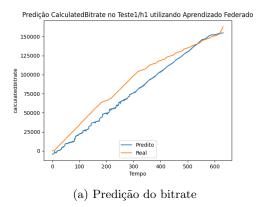
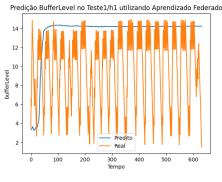


Figure 3: Ilustração da topologia de rede do Experimento I

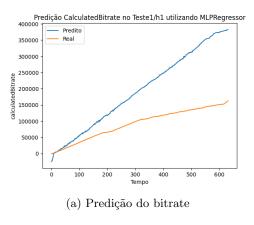
No primeiro experimento foram utilizados os hosts h2 e h3 para treino e o host h1 para teste. A Figura 4 apresenta o plot do resultado obtido utilizando aprendizado federado. A Figura 5 apresenta o plot do resultado obtido utilizando rede neural artificial (MLP). A Figura 6 apresenta o plot do resultado obtido utilizando floresta aleatória (RandomForest).

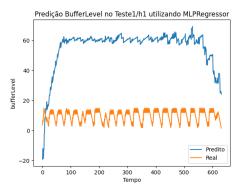




(b) Predição do buffer level

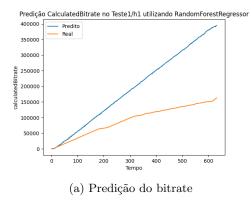
Figure 4: Gráficos de comparação entre os valores preditos e os valores reais no Experimento I utilizando aprendizado federado





(b) Predição do buffer level

Figure 5: Gráficos de comparação entre os valores preditos e os valores reais no Experimento I utilizando MLP



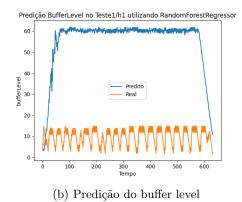


Figure 6: Gráficos de comparação entre os valores preditos e os valores reais no Experimento I utilizando RandomForest

3.3 Experimento II

A Figura 7 ilustra o cenário da topologia de rede.

No primeiro experimento foram utilizados os hosts h1 e h2 para treino e o host h3 para teste. A Figura 8 apresenta o plot do resultado obtido utilizando aprendizado federado. A Figura 9 apresenta o plot do resultado obtido utilizando rede neural artificial (MLP). A Figura 10 apresenta o plot do resultado obtido utilizando floresta aleatória (RandomForest).

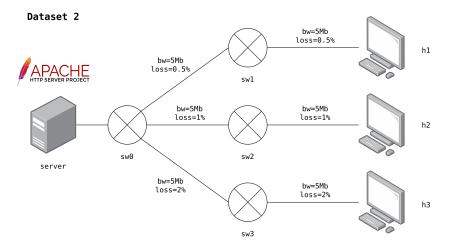


Figure 7: Ilustração da topologia de rede do Experimento II

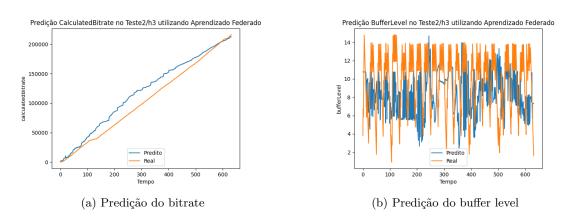


Figure 8: Gráficos de comparação entre os valores preditos e os valores reais no Experimento II utilizando aprendizado federado.

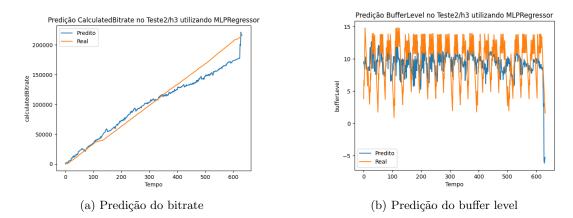
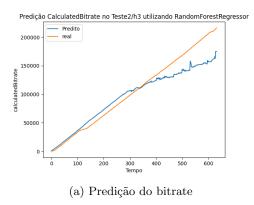
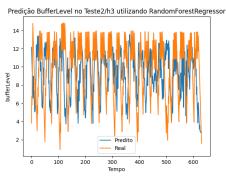


Figure 9: Gráficos de comparação entre os valores preditos e os valores reais no Experimento II utilizando MLP $\,$





(b) Predição do buffer level

Figure 10: Gráficos de comparação entre os valores preditos e os valores reais no Experimento II utilizando RandomForest

3.4 Experimento III

A Figura 11 ilustra o cenário da topologia de rede.

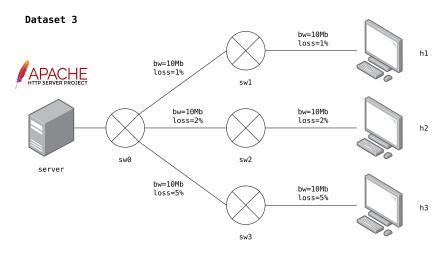
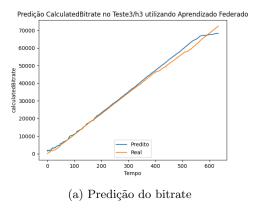
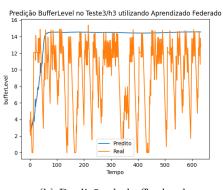


Figure 11: Ilustração da topologia de rede do Experimento III

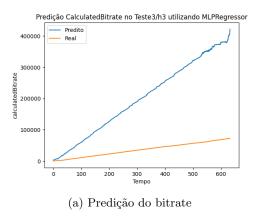
No primeiro experimento foram utilizados os hosts h1 e h2 para treino e o host h3 para teste. A Figura 12 apresenta o plot do resultado obtido utilizando aprendizado federado. A Figura 13 apresenta o plot do resultado obtido utilizando rede neural artificial (MLP). A Figura 14 apresenta o plot do resultado obtido utilizando floresta aleatória (RandomForest)

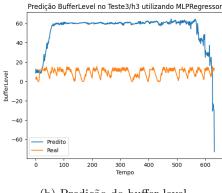




(b) Predição do buffer level

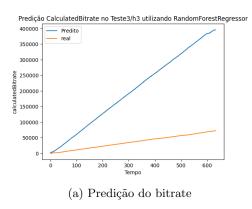
Figure 12: Gráficos de comparação entre os valores preditos e os valores reais no Experimento III utilizando aprendizado federado

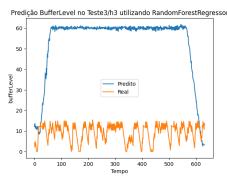




(b) Predição do buffer level

Figure 13: Gráficos de comparação entre os valores preditos e os valores reais no Experimento III utilizando MLP





(b) Predição do buffer level

Figure 14: Gráficos de comparação entre os valores preditos e os valores reais no Experimento III utilizando RandomForest

3.5 Experimento IV

A Figura 15 ilustra o cenário da topologia de rede.

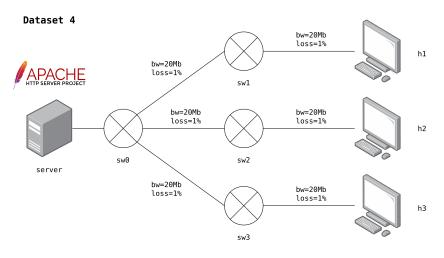
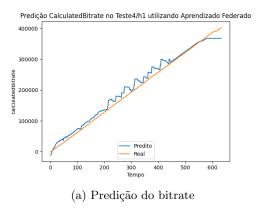


Figure 15: Ilustração da topologia de rede do Experimento IV

No primeiro experimento foram utilizados os hosts h2 e h3 para treino e o host h1 para teste. A Figura 16 apresenta o plot do resultado obtido utilizando aprendizado federado. A Figura 17 apresenta o plot do resultado obtido utilizando rede neural artificial (MLP). A Figura 18 apresenta o plot do resultado obtido utilizando floresta aleatória (RandomForest)



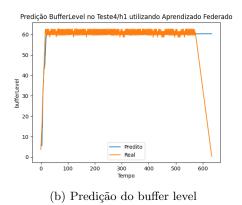
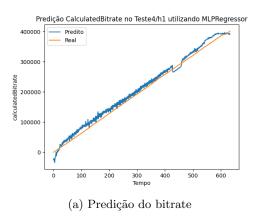


Figure 16: Gráficos de comparação entre os valores preditos e os valores reais no Experimento IV utilizando aprendizado federado



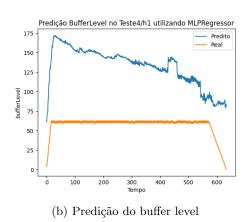
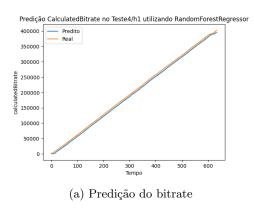


Figure 17: Gráficos de comparação entre os valores preditos e os valores reais no Experimento IV utilizando MLP



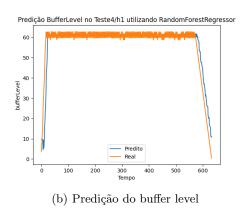


Figure 18: Gráficos de comparação entre os valores preditos e os valores reais no Experimento IV utilizando RandomForest

3.6 Discussões Gerais

Observando os resultados apresentados, pode-se notar que os modelos de aprendizado de máquina tiveram dificuldade, principalmente, na predição do bitrate e buffer level dos Experimentos I e III. Acreditamos que isso se deve ao fato de que há falta de diversidade nos dados de treino utilizados, *i.e.*, como só foram utilizados dois hosts para teste, e devido a grande diferença do cenário entre os *links* de rede dos hosts, o modelo pode não ter generalizado o bastante para fazer predição em um *link* com condições tão diferentes da que foram vistos no cenário de treino.

No Experimento II, que era um pouco mais homogêneo quanto as condições dos *links*, os resultados obtidos foram significativamente melhores que nos Experimentos I e III. Esse resultado corrobora com a hipótese levantada quanto a diversidade da base de treino.

Para verificar isso de forma melhor, o Experimento IV foi feito utilizando os mesmos *links*. Ainda sim, os modelos baseados em redes neurais artificiais (aprendizado federado e MLP) tiveram algum problema. Acredita-se que isso pode ser devido a falta de estudo profundo dos hiperparâmetros utilizados no modelo, principalmente quanto a qual otimizador foi utilizado. No entanto, o modelo de floresta aleatória conseguiu um resultado quase perfeito.

Apesar de alguns erros, acreditamos que os resultados obtidos mostraram tendência positiva, com necessidade de ajuste fino em algumas áreas para que os resultados fossem melhores.

4 Conclusão

Neste trabalho, foi proposto a predição de QoS de um serviço de streaming utilizando aprendizado federado. Esse cenário foi simulado utilizando o Mininet-Wifi. Para predição do QoS, além do modelo de aprendizado federado, também foram utilizados métodos clássicos de aprendizado de máquina como redes neurais artificiais e floresta aleatória.

A topologia de rede foi implementada com sucesso no Mininet-Wifi. O servidor Apache2 foi utilizado para disponibilização do vídeo para transmissão, e um *player* de vídeo baseado em MPEG-DASH foi utilizado para transmissão nos clientes. A partir desse cenário, quatro experimentos foram simulados variando a *bandwidth* e *loss* dos *links* da rede.

Os resultados obtidos nos experimentos tiveram alguns problemas, que acredita-se ser devido, principalmente, a falta de diversidade dos dados de treino. Como se trata de uma simulação em um cenário limitado, isso pode ter dificultado um melhor resultado dos modelos de aprendizado de máquina. Apesar disso, alguns resultados foram bem promissores, com boa predição do bitrate e buffer level.

Por fim, como trabalhos futuros pode-se comentar: (1) a implementação da topologia de rede utilizando WiFi; (2) estudo decomo aumentar a diversidade da base de treino, com um maior número de hosts, por exemplo; (3) captura de dados do servidor e não somente dos switches; (4) estudo de hiperparâmetros dos modelos de aprendizado de máquina, como, por exemplo, o otimizador utilizado nos modelos baseados em redes neurais artificiais; (4) necessidade de pensar em uma melhor forma de fazer a divisão de dados de treino e teste, também relacionado ao problema da diversidade dos dados.

References

- [1] Big buck bunny. https://peach.blender.org/download/.
- [2] Mininet wifi. https://mininet-wifi.github.io/.
- [3] Helio Cunha Neto, Diogo Menezes, and Natalia Fernandes. Privacidade do Usuário em Aprendizado Colaborativo: Federated Learning, da Teoria à Prática. 10 2020.
- [4] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-Efficient Learning of Deep Networks from Decentralized Data. In Aarti Singh and Jerry Zhu, editors, *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pages 1273–1282. PMLR, 20–22 Apr 2017.