CLASSIFICAÇÃO DO TRÁFEGO DE REDES CORPORATIVAS: UMA ABORDAGEM COM REDES NEURAIS ARTIFICIAIS

Deivid Weslley Pereira da Silva[1], Luiz Fernando Segato dos Santos[1]
Instituto Federal de Mato Grosso do Sul — Aquidauana-MS
deividweslps@gmail.com, luiz.santos@ifms.edu.br

Resumo

Este artigo aborda a importância da Qualidade de Serviço na gestão do tráfego de dados em redes, especialmente para garantir a qualidade de serviços sensíveis à latência. No entanto, com o crescimento exponencial do tráfego de dados e a complexidade das redes, manter a Qualidade de Serviço tornou-se um desafio. Uma solução promissora é a integração da inteligência artificial, especificamente redes neurais convolucionais, como a Resnet50. Este estudo descreve a coleta e processamento de dados, a implementação do método, a arquitetura neural, o treinamento da rede e os resultados alcançados. Demonstrando como a integração da inteligência artificial pode ser um passo significativo para melhorar o desempenho e a qualidade do serviço nas redes, o artigo ressalta a importância dessa abordagem em um cenário de redes em constante mudança e crescente demanda por largura de banda.

Palavras-chave: qualidade de serviço, classificação de tráfego, redes neurais convolucionais.

Introdução

Com o crescimento massivo de tráfego de dados hoje em dia e crescimento dos serviços veio desafios para gerenciar a qualidade desses serviços e a entrega eficaz desses dados é importante e muito utilizada hoje em dia, tem um papel importante no gerenciamento de tráfego pela rede e faz com que serviços importantes como o *Voice over IP* (VoIP) tenha uma boa qualidade de execução e não sofra perda de pacotes ou tenha latência muito alta, e a largura de banda ideal. As principais medições desta qualidade incluem fatores que afetam a qualidade das chamadas, como variação no atraso (conhecida como *jitter*), latência e perda de pacotes.

A Qualidade de Serviço, do inglês *Quality of Service* (QoS), é uma tecnologia presente em roteadores para garantir ao usuário maior controle sobre sua rede *Wi-Fi*. Por meio da ferramenta é possível determinar quais dispositivos e serviços terão maior prioridade de conexão. O recurso é interessante para quem precisa racionalizar a Internet, ou simplesmente precisa dar preferência para dispositivos que reproduzem vídeos em *streaming*, jogos *online*, entre outros tipos de uso. (GARRETT, 2018).

Para auxiliar esta demanda crescente, a integração da inteligência artificial na gestão do tráfego é uma solução promissora, por aumentar significativamente a qualidade do tráfego dos dados. Apesar dos avanços tecnológicos, garantir a qualidade do serviço nas redes de comunicações continua a ser um desafio complexo.

As exigências de largura de banda e a diversidade de aplicações e dispositivos conectados se tornam um gargalo para a infraestrutura de rede existente. Além disso, o tráfego de dados é altamente dinâmico, com demanda em constante mudança, dificultando que as operadoras de rede mantenham uma qualidade de serviço consistente.

Para este trabalho foi utilizado a técnica *Gramian Angular Field* (GAF) de conversão de números em imagens que será explicado melhor no parágrafo abaixo, pois o intuito deste trabalho é utilizar imagens para o treinamento da rede neural convolucional.

A técnica GAF resulta em uma imagem obtida de uma série temporal, representando algum tipo de correlação temporal entre cada par de valores da série temporal. Dois métodos estão disponíveis: *Gramian angular summation field* (GASF) e *Gramian angular difference field* (GADF). (PYTS, 2022).

As redes neurais convolucionais, do inglês *Convolutional Neural Networks* (CNN), se utilizam de uma arquitetura especial que é particularmente bem adequada para classificar imagens. O uso dessa arquitetura torna as redes convolucionais rápidas de treinar, o que é vantajoso para trabalhar com redes profundas. (PERES, 2021).

Este artigo apresenta como esta integração da inteligência artificial ao QoS pode ser um passo significativo na melhoria do rendimento e na garantia da qualidade do serviço em ambientes de redes utilizando CNNs como a Resnet50. Nas próximas seções serão apresentados como foram coletados os dados, tratamento dos dados, a implementação do método, a criação da arquitetura neural, treinamento da rede e por fim os resultados.

Metodologia

Neste projeto foi abordada uma técnica específica e muito importante para os dados utilizados no treinamento da

arquitetura chamada GAF, técnica essa que consiste em transformar dados numéricos em imagens GADF para fazer o treinamento utilizando CNNs. Para este projeto foi utilizado a CNN Resnet50, uma rede convolucional. Foi escolhida esta rede neural, pois neste trabalho o intuito é utilizar imagens para fazer o treinamento, por ser uma rede preparada para classificar imagens.

O Wireshark é um analisador de pacotes que, para tanto, tem capacidade de interceptar "todos" os pacotes que trafegam nas redes e de exibir detalhes das suas informações de controle (cabeçalhos) e conteúdo (*payload*). (HENRIQUE, 2018).

Para fazer a coleta de dados foi utilizado o software Wireshark onde foram obtidos pacotes de *streaming* e no site do Wireshark estão disponíveis gratuitamente dados de VoIP que foram utilizados neste projeto, então temos dois tipos de dados "*Streaming*" e "VoIP" para fazer uma rede neural binária. Após isso teve de ser coletado os metadados que se referem às informações adicionais associadas aos dados da rede que estão sendo analisados. Os metadados analisados para este projeto foram: *Timestamp, Source, Destination, Protocol_UDP, Protocol_TCP, Length, Jitter, Bandwidth, Throughput, Latency, BytesPerSecond, PacketsPerSecond, PacketSize*. Para a coleta destes metadados foi feito um script que coleta estas métricas automaticamente dos arquivos pkt referentes aos dados de "VoIP" e "*Streaming*" e adiciona ao arquivo csv obtendo assim dados necessários para o treino da rede neural. Logo após foi feita a conversão desses dados numéricos para imagens com a técnica GAF.

Utilizando a técnica GAF foi feita a conversão desses dados numéricos. Podemos conferir a seguir os metadados e o resultado GAF nas Figuras 1 e 2, respectivamente. Essas imagens resultantes são pequenas e de baixa resolução, contendo 13 x 13 pixels.

	Timestamp	Source	Destination	Protocol_UDP	Protocol_TCP	Length	Jitter	Bandwidth	Throughput	Latency	BytesPerSecond	PacketsPerSecond	PacketSize
0	1.480173e+09	167772692	167772687	1	0	508	0.000000	0.000000e+00	4064.000000	0.000000	0.000000e+00	0.000000	508
1	1.480173e+09	167772687	167772692	1	0	331	0.000173	3.679769e+07	6710.839025	0.000173	3.679769e+07	5780.346821	331
2	1.480173e+09	167772687	167772692	1	0	47	0.002386	2.769832e+06	7069.908105	0.002386	2.769832e+06	781.555295	47
3	1.480173e+09	167772687	167772692	1	0	1111	0.001597	3.844081e+06	15909.678545	0.001597	3.844081e+06	721.847931	1111
4	1.480173e+09	167772692	167772687	1	0	357	0.000128	4.395892e+06	18751.667855	0.000128	4.395892e+06	933.706816	357
...	...	...	...	...	...	...	...	...	...	...	...	...	...
795	1.480173e+09	167772687	167772692	1	0	114	0.020014	4.383565e+04	41222.573271	0.020014	4.383565e+04	50.394698	114
796	1.480173e+09	167772687	167772692	1	0	114	0.019998	4.383789e+04	41227.790808	0.019998	4.383789e+04	50.394205	114
797	1.480173e+09	167772687	167772692	1	0	114	0.020007	4.384010e+04	41232.973867	0.020007	4.384010e+04	50.393684	114
798	1.480173e+09	167772687	167772692	1	0	114	0.019987	4.384236e+04	41238.193696	0.019987	4.384236e+04	50.393228	114
799	1.480173e+09	167772687	167772692	1	0	114	0.020004	4.384457e+04	41243.359350	0.020004	4.384457e+04	50.392719	114

800 rows × 13 columns

Figura 1. Dataset em forma de tabela.

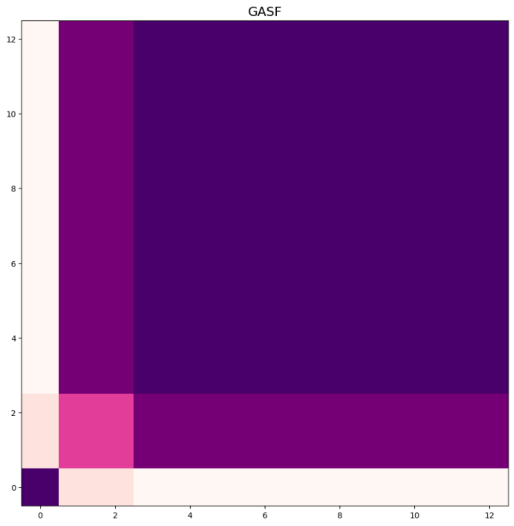

Figura 2. Imagem resultante do processo GAF.

O uso desta técnica resultou em 4194 imagens, cada imagem é equivalente a uma linha dos arquivos csv. Na Tabela 1 podemos observar a quantidade imagens distribuída para cada classe.

Tabela 1. Quantidade de amostras para cada classe.

Classe	Amostras
Streaming	2374
VoIP	1820

Tendo os dados em imagens foi dado início a criação da arquitetura da rede neural. Para este treinamento utilizamos a Resnet50, então as camadas iniciais foram congeladas e adicionadas algumas camadas dividindo em duas partes, a primeira contendo camadas *pooling* e a segunda parte contendo as camadas densas, porque foram anteriormente feitos testes e foi visto que seria necessário adicioná-las para ter uma arquitetura mais robusta e a segunda parte foram as camadas densas e de *dropout* para regularização do modelo. Na Figura 3 a seguir temos o exemplo do código dessas camadas.

```
# Camadas convolucionais e de max pooling
x = tf.keras.layers.Conv2D(32, (3, 3), input_shape = (64, 64, 3), activation='relu', padding='same')(x)
x = tf.keras.layers.MaxPooling2D((2, 2))(x)
x = tf.keras.layers.Conv2D(64, (3, 3), activation='relu', padding='same')(x)
x = tf.keras.layers.MaxPooling2D((2, 2))(x)
x = tf.keras.layers.Dropout(0.2)(x)  #Dropout para regularização

# Camada densa
x = tf.keras.layers.GlobalAveragePooling2D()(x)
x = tf.keras.layers.Flatten()(x)
x = tf.keras.layers.Dense(2048, activation='relu')(x)
x = tf.keras.layers.Dropout(0.2)(x)  #Dropout para regularização
x = tf.keras.layers.Dense(1024, activation='relu')(x)
x = tf.keras.layers.Dropout(0.2)(x)  #Dropout para regularização
x = tf.keras.layers.Dense(512, activation='relu')(x)
x = tf.keras.layers.Dropout(0.2)(x)  #Dropout para regularização

preds = tf.keras.layers.Dense(1, activation='sigmoid')(x)
```

Figura 3. Arquitetura neural.

Foram realizados quatro treinos com divisão das imagens para validação e treinamento diferentes, o primeiro com 20% do total para validação, o segundo com 25%, o terceiro com 30% e o quarto com 35%. Isto foi feito para saber se mudando esta porcentagem houvesse uma melhora nos resultados do treino. Então foram feitos os treinos com 300 de épocas, isso será mostrado com mais detalhes na seção resultados.

Resultados e Discussão

Como foi dito acima, durante o processo de treinamento foi necessário fazer algumas adaptações mudando a porcentagem de validação e adicionando algumas camadas extras para regularização do modelo.

Na Figura 4 podemos ver os resultados do primeiro treinamento com 20% de validação e três camadas densas, sendo uma com 2048 neurônios, outra com 1024 e outra com 512, e as mesmas camadas de pooling da Figura 3 menos a camada de *dropout*. Na Tabela 2 estão descritas as médias obtidas do modelo. A Figura 4 a seguir mostra os resultados em formato de gráfico durante as épocas.

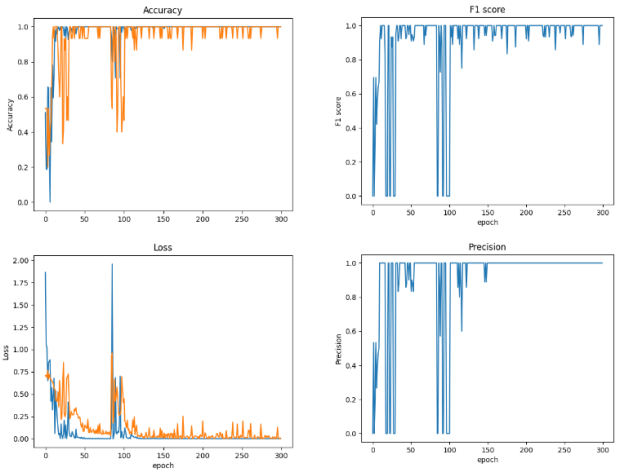

Figura 4. Gráficos do primeiro treinamento.

Tabela 2. Médias do primeiro treinamento.

Média	Validação
Accuracy	0.943
Loss	0.135
F1_score	0.910
Precision	0.912

Analisando estes gráficos podemos perceber alguns picos durante o treinamento, tendo isso em vista foi adicionado algumas camadas de *dropout* que reduziu esses picos como podemos conferir na Figura 5 e suas respectivas médias na Tabela 3.

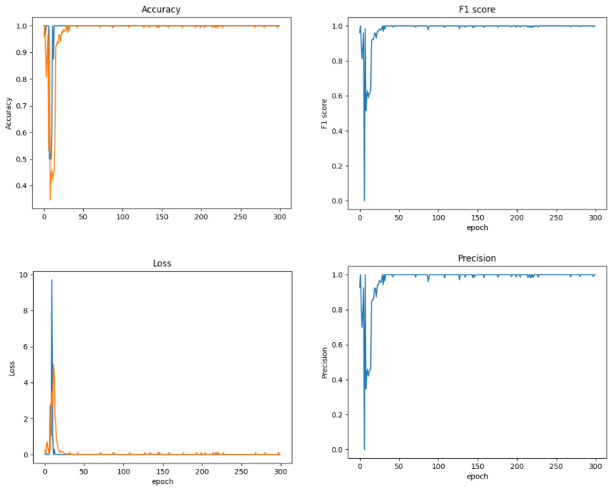

Figura 5. Gráficos do segundo treinamento.

Tabela 3. Médias do segundo treinamento.

Média	Validação
Accuracy	0.981
Loss	0.112
F1_score	0.983
Precision	0.975

Neste terceiro treino foi mudado apenas o split de validação, a porcentagem foi alterada de 20% para 25%. Os resultados estão representados na Figura 6. Na tabela 4 podemos conferir o resultado das médias.

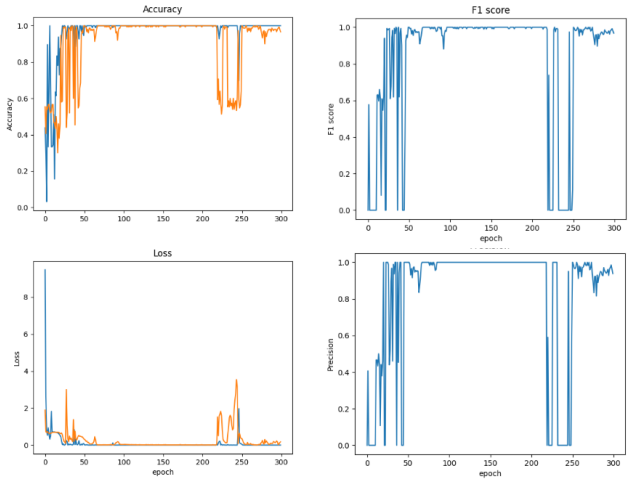

Figura 6. Gráficos do terceiro treinamento.

Tabela 4. Médias do primeiro treinamento.

Média	Validação
Accuracy	0.908
Loss	0.254
F1_score	0.838
Precision	0.832

Para o quarto treino alterou-se a porcentagem de validação para 30% representado na Figura 7 e as médias do modelo na Tabela 5.

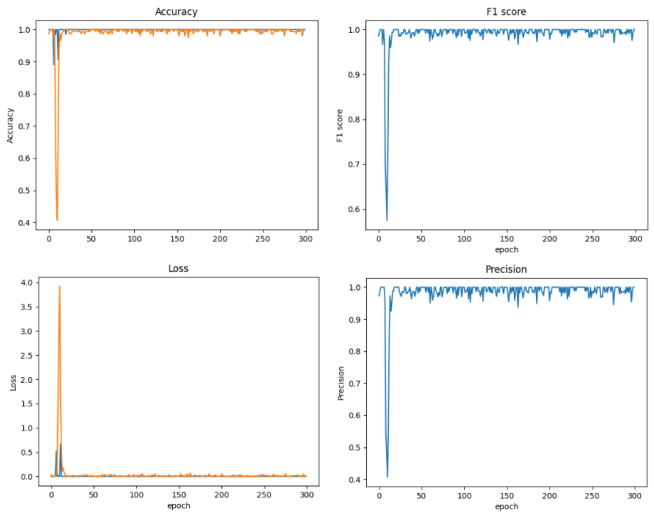

Figura 7. Gráficos do quarto treinamento.

Tabela 5. Médias do primeiro treinamento.

Média	Validação
Accuracy	0.990
Loss	0.047
F1_score	0.990
Precision	0.983

E por fim, no último treinamento a porcentagem foi alterada para 35%. Os resultados estão apresentados na Figura 8 e as médias na Tabela 6.

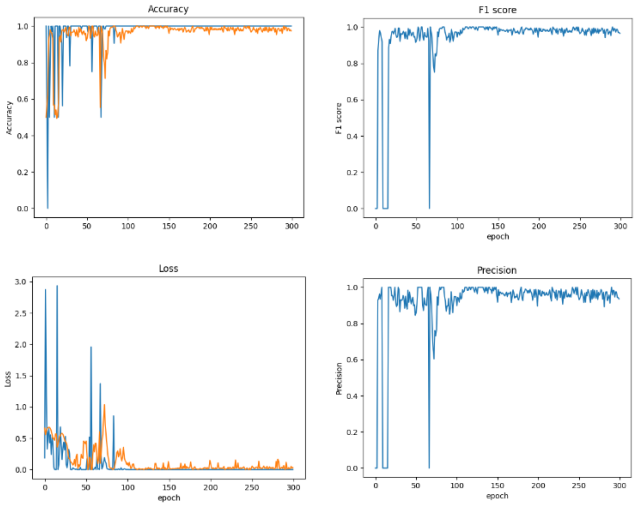

Figura 8. Gráficos do quinto treinamento.

Tabela 6. Médias do primeiro treinamento.

Média	Validação
Accuracy	0.960
Loss	0.126
F1_score	0.936
Precision	0.919

Considerações Finais

Tendo em vista a evolução do projeto, pode-se dizer que obteve excelentes resultados durante o desenvolvimento do mesmo e atingiu grande parte dos objetivos propostos.

Em projetos futuros ainda haverá outros testes a serem realizados, como, por exemplo, alterar a classificação binária de dois elementos para três ou mais e também um aumento nos dados. E com outros trabalhos pode ser realizado implementação desta rede em um cenário real.

Agradecimentos

Ao IFMS pela bolsa concedida para execução deste projeto.

Referências

GARRETT, Felipe. **O que é QoS? Entenda para que serve a tecnologia em roteadores.** Techtudo. 2018. Disponível em: <https://www.techtudo.com.br/noticias/2018/07/o-que-e-qos-entenda-para-que-serve-a-tecnologia-em-roteadores.ghtml> Acesso em: 20 de setembro. 2023.

PYTS. **Single Gramian angular field.** [S.I] [2022]. Disponível em: <https://pyts.readthedocs.io/en/stable/auto_examples/image/plot_single_gaf.html> Acesso em: 20 de setembro. 2023.

PERES, Lucas. **Aprenda a Criar e Treinar Uma Rede Neural Convolucional (CNN).** Insight lab. 2021. Disponível em: <https://insightlab.ufc.br/aprenda-a-criar-e-treinar-uma-rede-neural-convolucional-cnn/#:~:text=As%20redes%20neurais%20convolucionais%20%28CNN%29%20se%20utilizam%20de,que%20%C3%A9%20vantajoso%20para%20trabalhar%20com%20redes%20profundas> Acesso em: 20 de setembro. 2023.

HENRIQUE, Samuel. **Wireshark na Análise de Tráfego e Protocolos em Redes. labCisco.** 2018. Disponível em: <http://labcisco.blogspot.com/2013/11/wireshark-na-analise-de-trafego-e.html#:~:text=O%20Wireshark%20%C3%A9%20um%20analisador,)%20e%20conte%C3%BAdo%20(payload)> Acesso em: 20 de setembro. 2023.

CLASSIFICATION OF CORPORATE NETWORK TRAFFIC: AN APPROACH WITH ARTIFICIAL NEURAL NETWORKS

Abstract: *This article addresses the importance of Quality of Service in managing data traffic in networks, especially to guarantee the quality of latency-sensitive services. However, with the exponential growth of data traffic and the complexity of networks, maintaining Quality of Service has become a challenge. One promising solution is the integration of artificial intelligence, specifically convolutional neural networks such as Resnet50. This study describes data collection and processing, method implementation, neural architecture, network training and the results achieved. Demonstrating how the integration of artificial intelligence can be a significant step towards improving performance and quality of service in networks, the article highlights the importance of this approach in a scenario of constantly changing networks and increasing demand for bandwidth.*

Keywords: Quality of service, traffic classification, convolutional neural networks.