CIÊNCIA DA COMPUTAÇÃO: PRINCÍPIOS FUNDAMENTAIS

VOLUME V

EDITORA CONHECIMENTO LIVRE

Frederico Celestino Barbosa

Ciência da computação: princípios fundamentais

5ª ed.

Piracanjuba-GO
Editora Conhecimento Livre
Piracanjuba-GO

5ª ed.

Dados Internacionais de Catalogação na Publicação (CIP)

Barbosa, Frederico Celestino
B238C Ciência da computação: princípios fundamentais

/ Frederico Celestino Barbosa. – Piracanjuba-GO

Editora Conhecimento Livre, 2022

140 f.: il
DOI: 10.37423/2022.edcl529
ISBN: 978-65-5367-151-5
Modo de acesso: World Wide Web
Incluir Bibliografia

1. tecnologia 2. informática 3. codificação 4. sistemas I. Barbosa, Frederico Celestino II. Título

CDU: 4

https://doi.org/10.37423/2022.edcl529

EDITORA CONHECIMENTO LIVRE

Corpo Editorial

Editora Conhecimento Livre
Piracanjuba-GO
2022

SUMÁRIO

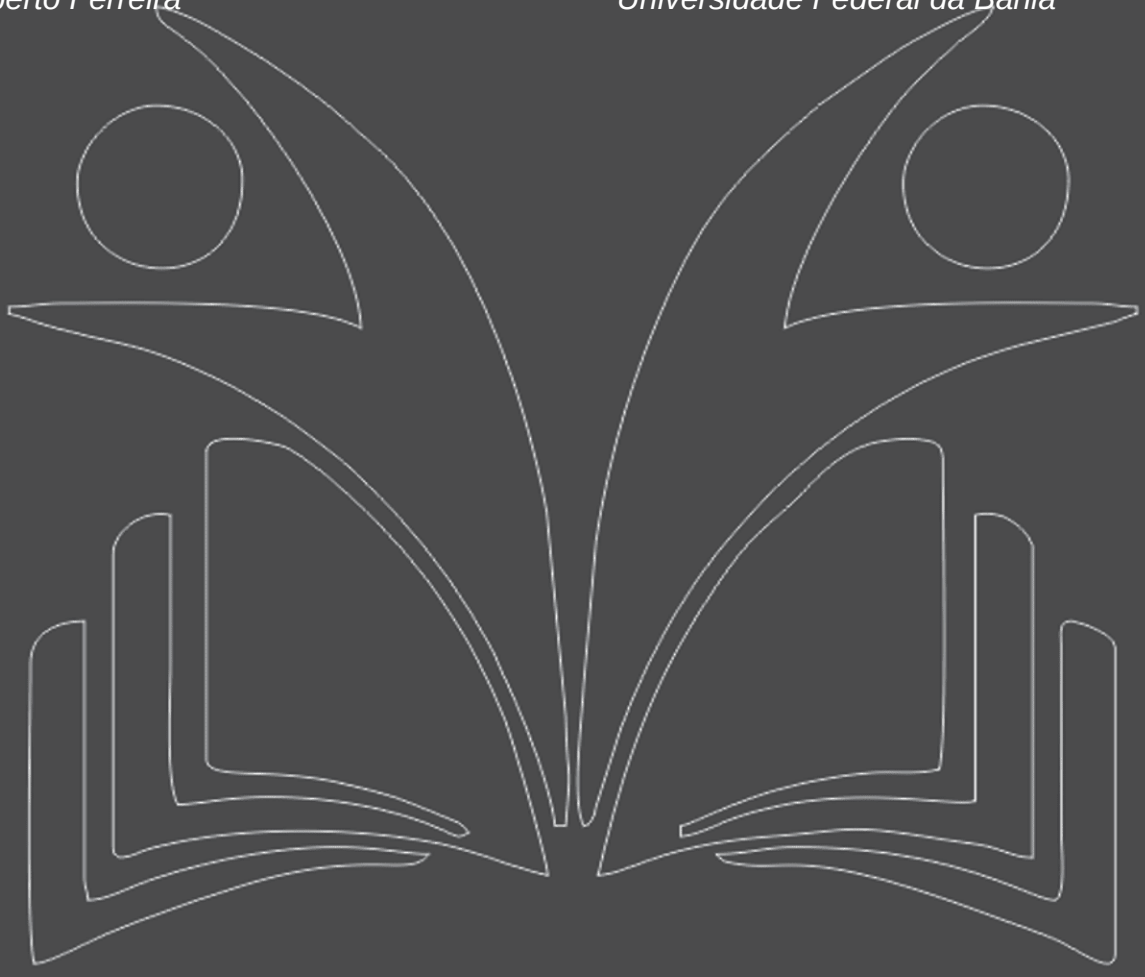
Capítulo 1

10.37423/220606146

RECONFIGURAÇÃO DE REDES DE DISTRIBUIÇÃO DE ENERGIA ELÉTRICA UTILIZANDO METAHEURÍSTICA COLÔNIA DE FORMIGAS MODIFICADA

Huilman Sanca Sanca — *Universidade Federal da Bahia*

Niraldo Roberto Ferreira — *Universidade Federal da Bahia*



Resumo. *Neste artigo apresenta-se um algoritmo colônia de formigas (ACO) modificada para resolver o problema de reconfiguração ótima de redes de distribuição de energia elétrica. Este problema de otimização é não linear e de natureza combinatória. Em cooperação as formigas descobrem os menores caminhos entre o formigueiro e uma fonte de alimento usando um mecanismo de comunicação indireta. Visando melhorar o desempenho do algoritmo foi dotado de uma técnica de elitismo no processo da convergência. O método foi aplicado a um alimentador de 69 barras.*

Palavras chaves. *Reconfiguração, colônia de formigas, elitismo, rede de distribuição, perdas ativas, fluxo de carga radial.*

1. INTRODUÇÃO

Os sistemas de distribuição de energia elétrica devem operar de forma confiável e econômica, respeitando os parâmetros de qualidade da energia. O problema da reconfiguração consiste em encontrar outra configuração radial que apresente o menor valor de perdas de potencia ativa do sistema melhorando o perfil de tensão nas barras. A reconfiguração de sistemas de distribuição de energia elétrica é levada a efeito operando- se chaves existentes de modo que a topologia da rede seja alterada para que haja transferência de cargas de um alimentador fortemente carregado para outro, relativamente, menos carregado.

Reconfiguração de redes é um problema de otimização combinatória não diferençável [Ching Tzong et al. 2005]. Devido ao elevado número de combinações necessárias, metaheurísticas têm sido empregadas para resolveˆ-lo. Diferentes métodos e técnicas de inteligência artificial têm sido aplicados a reconfiguração de redes. Em [Lin et al. 2000], utiliza algoritmos genéticos. Em [Jeon et al. 2002], [Parada et al. 2004], os autores propuseram a aplicação do algoritmo de têmpera simulada (simulated annealing algorithm). Em [Olamaei et al. 2007], [Lu et al. 2009], obtem-se a solução do problema mediante o algoritmo de nuvem de partículas (Particle Swarm Optimization).

O algoritmo colônia de formigas foi aplicada pela primeira vez ao problema clássico do caixeiro viajante, Traveling Salesman Problem (TSP) [Dorigo and Stutzle 2004]. O algoritmo colônia de formigas é inspirado no comportamento das formigas reais, em particular, como e´ conhecido, as formigas reais são capazes de encontrar o caminho mais curto a partir do formigueiro para a fonte de alimento sem a utilização de sinais visuais, o meio de comunicação das formigas ocorre mediante uma substância química depositada por elas chamada de feromônio.

Neste artigo, analisa-se um método ACO modificado que emprega regras de atualização global e local de feromônio, para efeito de melhoramento de seu desempenho. Com a finalidade de demonstrar a eficácia do algoritmo é aplicada para reconfigurar um alimentador de 69 barras apresentado em [Chiang and Jean-Jumeau 1990], e que consta de 69 barras e 73 ligações, fornecendo bons resultados na procura da melhor configuração da rede com menores perdas de potência ativa.

2. FORMULAÇÃO DO PROBLEMA

2.1. FORMULAÇÃO DO PROBLEMA DE RECONFIGURAÇÃO DE REDES PARA O ACO ELITISTA

A reconfiguração de redes pode ser formulado como um problema de otimização não linear mista, de variáveis inteiras e reais, cuja expressão é:

$$Minimize f(x) = P_{T,Perdas} = \sum_{i=1}^{N_B-1} P_{perdas}(i, i+1) \qquad (1)$$

Sujeito ás restrições:

1. de fluxo de carga;

2. limites de corrente nos trechos:

$$|I_j| \leq I_{j,max}; \forall j, j \in N_T \qquad (2)$$

3. CONFIGURAÇÃO RADIAL DA REDE.

sendo, $P_{T,perdas}$ a perda de potência total do sistema, I_i e $I_{i,max}$ são a amplitude da corrente e o limite máximo de corrente em cada trecho *i*, respectivamente, N_B é o número de barras e N_T e´ o número de trechos do sistema.

O problema de optimização com restrições expresso em (1), pode ser convertido em um problema de optimização irrestrito cuja função objetivo incorpore a restrição de corrente máxima nos trechos [Lin et al. 2000].

$$Minimize f(x) = \sum_{i=1}^{N_B-1} P_{perdas}(i, i+1) + \sum_{j=1}^{N_T} \lambda_I (I_j - I_{j,max})^2 \qquad (3)$$

sendo λI um fator de penalidade com respeito a` corrente admissível no trecho.

2.2 CÁLCULO DAS PERDAS

O cálculo das perdas de potência é parte do problema do fluxo de carga, que aqui é resolvido empregando-se o método da soma de potência (MSP). O MSP, é um método iterativo nas variáveis de perdas de potência ativa e reativa do tipo *forward - backward* [Haque 2000]. Os valores absolutos

das tensões das barras são calculadas sequencial- mente no sentido das subestações para as barras terminais (*forward*). As potencias dos trechos são calculados sequencialmente no sentido das barras terminais para as subestações (*backward*). Para o cálculo de um trecho (*i,i+1*). Referido a` Figura 1, podem-se escrever estas equações como em [Haque 2000]:

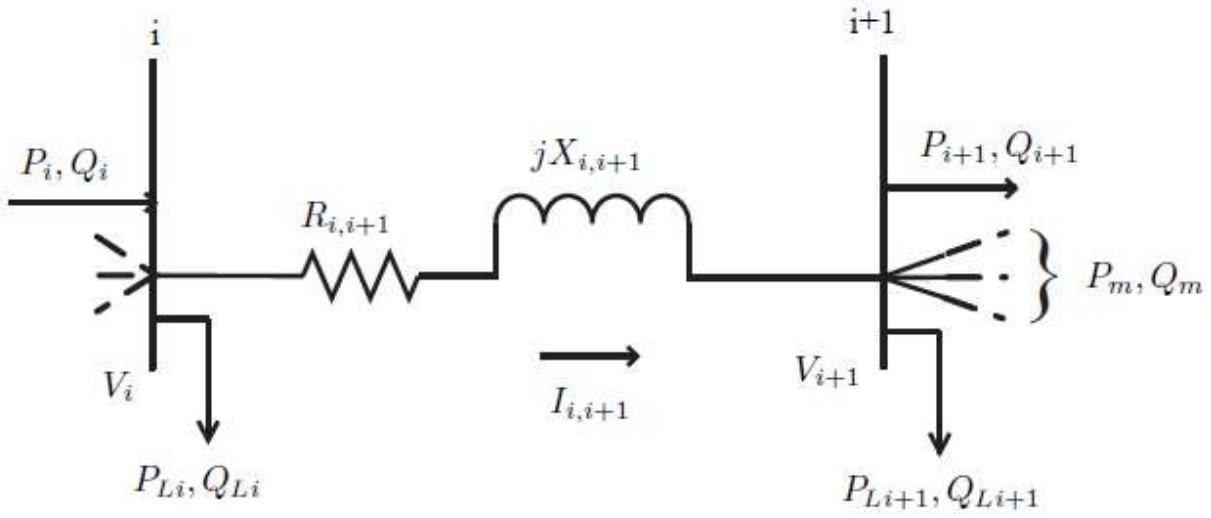


Figura 1. Trecho de uma rede de distribuição simples.

$$P_i = P_{i+1} + P_{Li+1} + R_{i,i+1}\left[\frac{P_i^2 + Q_i^2}{|V_i|^2}\right] \tag{4}$$

$$Q_i = Q_{i+1} + Q_{Li+1} + X_{i,i+1}\left[\frac{P_i^2 + Q_i^2}{|V_i|^2}\right] \tag{5}$$

$$|V_{i+1}|^2 = |V_i|^2 - 2(R_{i,i+1}P_i + X_{i,i+1}Q_i) + (R_{i,i+1}^2 + X_{i,i+1}^2)\frac{(P_i^2 + Q_i^2)}{|V_i|^2} \tag{6}$$

sendo P_i e Q_i os fluxos de potência ativa e reativa no trecho entre as barras i e $i + 1$, respectivamente, P_{Li} e Q_{Li} as potencias ativa e reativa das cargas instaladas na barra i, respectivamente. A resistência e a reatância do trecho entre as barras i e $i + 1$, são indicadas por $R_{i,i+1}$ e $X_{i,i+1}$, respectivamente.

Uma expressão alternativa á (6) é:

$$|V_{i+1}| = \sqrt{A + \sqrt{A^2 - B}}, \tag{7}$$

sendo:

$$A = \frac{|V_i|^2}{2} - (R_{i,i+1}P_i + X_{i,i+1}Q_i)$$

E

$$B = (R_{i,i+1}^2 + X_{i,i+1}^2)(P_i^2 + Q_i^2).$$

A perda de potência ativa no trecho é dada por:

$$P_{perdas}(i, i+1) = R_{i,i+1}\frac{P_i^2 + Q_i^2}{|V_i|^2}. \quad (8)$$

A perda de potência ativa total do sistema $P_{T,Perdas}$, é potência ativa de todos os trechos do sistema:

$$P_{T,Perdas} = \sum_{i=1}^{N_B-1} P_{perdas}(i, i+1). \quad (9)$$

3. ALGORITMO COLONIA DE FORMIGAS

3.1 O COMPORTAMENTO DA COLÔNIA DE FORMIGAS

O algoritmo colônia de formigas, ACO, é uma metaheurística inspirada nas formigas reais [Dorigo and Stutzle 2004], [Ding and Loparo 2012]. Na Figura 2 apresenta-se o processo de procura do caminho mais curto entre o formigueiro ”B” e a fonte de alimento ”A”.

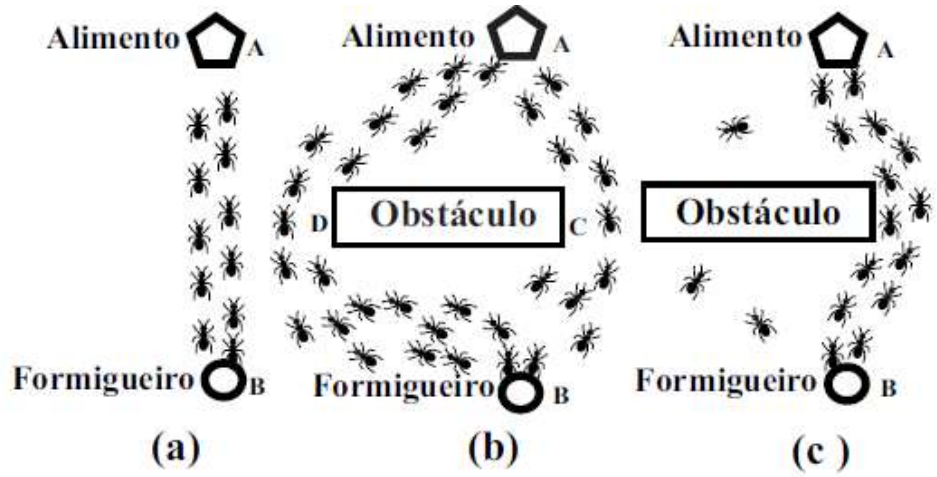


Figura 2. Exemplo de uma colônia de formigas real.

Na busca por alimento as formigas inicialmente se movimentam sem orientação tomando decisões com base em critérios individuais. Mais tarde, as formigas que escolheram o caminho mais curto entre o formigueiro e a fonte de alimento completarão suas expedições mais rapidamente. Isto fara´ que mais formigas escolham o menor caminho devido á maior concentração de feromônio. No fim, a maioria das formigas irão escolher o mesmo caminho devido á atualização constante do feromônio. Este comportamento mostra o paradigma fundamental do algoritmo de busca por colônia de formigas.

3.2. ESCOLHA PSEUDOALEATÓRIA DAS LIGAÇÕES

Trazendo o modo natural das formigas descobrir e coletar alimentos para resolver o problema da reconfiguração de redes pode-se imaginar o seguinte: a formiga escolhe a barra seguinte a ser visitada com base em seu próprio conhecimento que tem da rede (resistência das ligações entre a barra onde ficou a formiga e as barras vizinhas a ela) e no conhecimento coletivo (quantidade de feromônio depositado nas ligações durante o percurso). O conhecimento coletivo e´ cumulativo, sendo alterado sempre que uma nova configuração radial se completa. A probabilidade de uma das ligações vizinhas vir a ser escolhida por uma formiga e´ dada pela expressão:

$$Prob_{(i,j)} = \begin{cases} \dfrac{[\tau_{(i,j)}]^{\alpha}[\eta_{(i,j)}]^{\beta}}{\sum_{m \in J_{K(i)}} [\tau_{(i,m)}]^{\alpha}[\eta_{(i,m)}]^{\beta}} & \text{se} \quad \forall j \in J_{K(i)}; \\ 0 \quad \text{se} \quad \forall j \notin J_{K(i)}, \end{cases} \tag{10}$$

Sendo τ a quantidade de feromônio na ligação escolhida (*i, j*), cuja resistência é $1/\eta_{(i,m)}$. α e β são os pesos do grau de atratividade e a visibilidade do feromônio respectivamente. $J_{K(i)}$ é o conjunto das ligações vizinhas que poderão ser visitadas pela formiga *K* que se encontra no no´ *i*.

3.3 REGRA DE ATUALIZAÇÃO LOCAL DO FEROMÔNIO

Logo após as formigas percorrerem cada trecho, entre as barras *i* e *j*, aplica-se a regra de atualização local do feromônio, conforme a seguinte expressão:

$$\tau_{(i,j)} = (1-\rho)\tau_{(i,j)} + \rho\tau_0 \tag{11}$$

sendo $\tau_{(i,j)}$ a carga de feromônio na ligação (*i, j*), τ_0 o valor inicial do feromônio e ρa taxa de evaporação do feromônio, $\rho \in [0, 1]$. Esta regra de atualização local serve para diminuir a

possibilidade de estagnação do processo e aumentar a probabilidade de encontrar uma melhor solução até o momento[Dorigo and Stutzle 2004].

3.4 REGRA DE ATUALIZAÇÃO GLOBAL DO FEROMÔNIO

Logo após uma configuração radial factível ser achada são calculadas as perdas de potência ativa da rede e a carga de feromônio nas ligações da melhor configuração é incrementada do seguinte modo:

$$\tau_{(i,j)} = (1 - \sigma)\tau_{(i,j)} + \sigma\triangle\tau \qquad (12)$$

sendo:

$$\triangle\tau = \frac{1}{F_{melhor}} \qquad (13)$$

e F_{melhor} é o valor da função objetivo da melhor solução achada ate´ o momento, σ é a taxa de evaporação da regra de atualização global, $\sigma \in [0, 1]$. O feromônio é uma substância volátil o suficiente para evitar que soluções antigas sejam demasiadamente persistentes. (1 − σ) modela a evaporação do feromônio.

4. APLICAÇÃO DO ALGORITMO PARA O PROBLEMA DE RECONFIGURAÇÃO DE REDES

Inicialmente todos os trechos têm a mesma quantidade de feromônio, todos os nós fontes estão ligados, portanto os nós de carga estão todos desligados, mas nenhuma ligação está ativada. No´ ligado é aquele onde esta´ a formiga no momento, ligação ativada e´ aquela que ja´ foi percorrida e ligação ativável é aquela ligação vizinha para a qual a formiga pode se deslocar. O algoritmo ACO Elitista proposto e´ aquele que se apresenta na Figura 3.

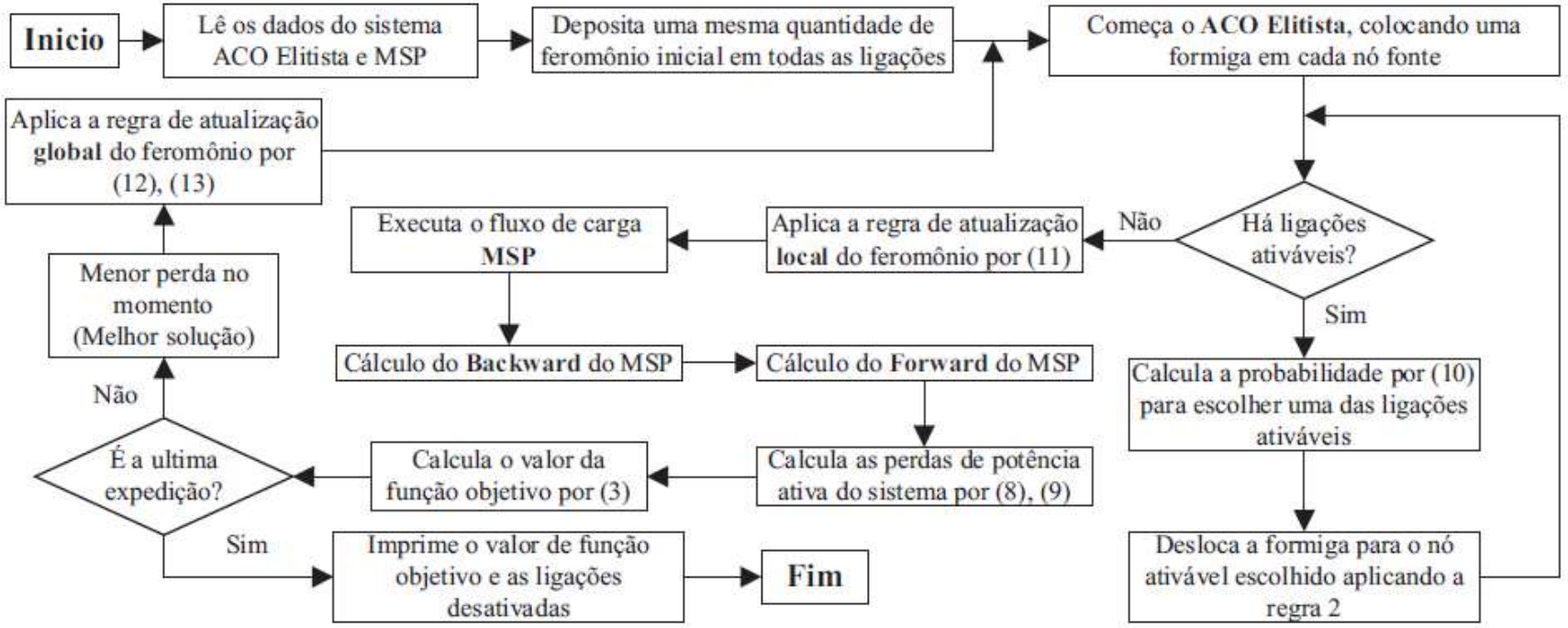


Figura 3. Fluxograma do ACO com Elitismo.

As formigas partem simultaneamente de forma aleatória de nós ligados respeitando as seguintes regras:

1. As formigas se deslocam exclusivamente por ligações ativáveis;
2. Quando uma formiga chega ao nó desligado da ligação ativável que tenha percorrido:

- este nó torna-se ligado e a ligação ativada;
- surge outra formiga para ocupar o nó originalmente ligado deixado por ela;

3. O percurso de uma formiga se completa quando ela não puder mais seguir por ligaçõess ativáveis
4. A expedição termina quando nenhuma formiga tiver mais mobilidade, ou seja, quando não houver nenhuma ligação ativável.

Durante uma iteração, apenas uma lista de barras a serem visitadas (barra ativáveis vizinhas ao nó ligado) é utilizada e escolhida de forma probabilística, atualiza-se a lista conforme os agentes caminham sobre a rede. Quando um agente se encontra sobre a barra i, seleciona a barra vizinha j (diretamente conectada a i) a ser visitada com base na concentração de feromônio sobre a linha (i,j) e no inverso da resistência desta, só são consideradas barras ativáveis aquelas barras que são diretamente conectadas a barra ligada (barra onde a formiga esta no momento).

5. TESTE E RESULTADOS

O algoritmo proposto foi implementado em Matlabr e testado com o alimentador de 69 barras da Figura 4, no qual há 5 chaves de interconexão e 73 chaves seccionadoras, originalmente 68 chaves estão fechadas (chaves de 1 a 68) e 5 abertas (chaves de 69 a 73).

Para esta configuração inicial as perdas ativas iniciais são de 20,9 kW e a tensão de linha na saída da subestação (barra 1) é de 12,66 kV. Os parâmetros utilizados no algoritmo são mostrados na Tabela 1.

A solução, que se mostra na Tabela 2, foi alcançada em 3,79 segundos utilizando- se um computador com processador Intel core i5-2410M de 2,3 GHz e 6 GB de RAM.

Tabela 1. Parâmetros usados para o teste do algoritmo

Parâmetros	Símbolo	Valor	Parâmetros	Símbolo	Valor
Alfa	α	0,1	Expedições		170
Beta	β	1,9	Sigma	σ	0,01
Rô	ρ	0,1	Feromônio inicial	τ_0	1
Tolerância do MSP	ε	10^{-3}			

A perda ativa total para a configuração inicial do sistema e´ de 20,9 kW, enquanto para a configuração obtida pelo algoritmo proposto elas foram de 9,35 kW, isto equivale a uma redução de 55,26%.

Fazendo-se uma comparação com os resultados publicados por [Chiang and Jean-Jumeau 1990] e [Abdelaziz et al. 2012], conforme mostra-se na Tabela 2, nota-se o bom desempenho do algoritmo proposto. Na Figura 5 apresenta-se

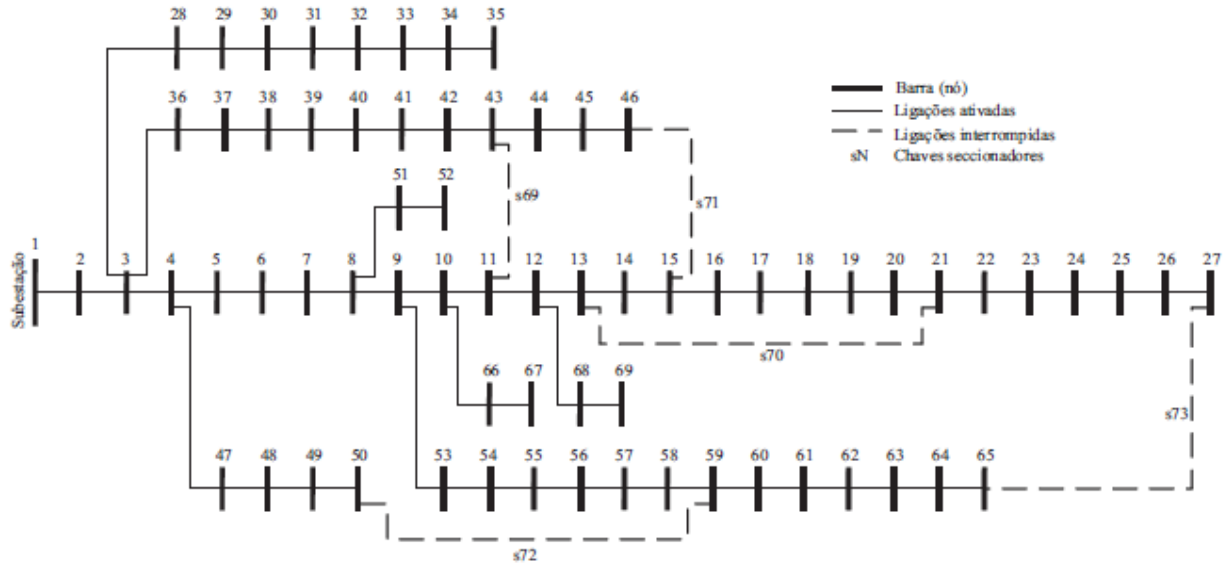


Figura 4. Sistema de distribuição de 69 barras e 73 ligações, [Chiang and Jean-Jumeau 1990].

Tabela 2. Resultados do teste

Configuração	Perdas finais (kW)	Redução (%)	Chaves desligadas
Inicial	20,9	- - -	s69, s70, s71, s72, s73
ACO Elitismo	9,35	55,26	s10, s57, s62, s70, s71
[Chiang and Jean-Jumeau 1990]	9,34	55,31	s15, s57, s62, s70, s71
[Abdelaziz et al. 2012]	9,355	54,30	s14, s58, s61, s69, s70

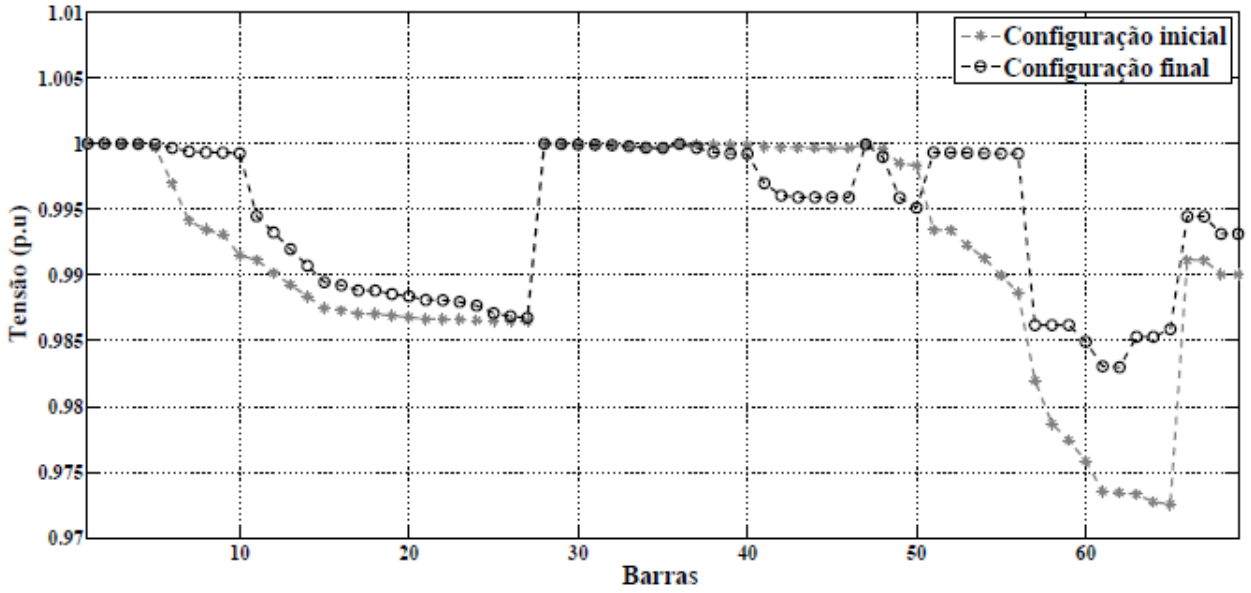


Figura 5. Comparação de tensões das barras para a configuração inicial e a configuração ótima encontrada.

Uma comparação do perfil das tensões em cada barra nas configurações inicial e final (ótima).

Observa-se a melhoria obtida no perfil de tensão decorrente da reconfiguração alcançada desativando-se as chaves seccionadoras s10, s57, s62, s70, s71 que equivalem aos trechos dos nós: 10-11, 57-58, 62-63, 13-21, 15-46.

6.CONCLUSÃO

Um algoritmo de formigas modificado (ACO Elitista) foi proposto neste artigo com o objetivo de encontrar uma configuração radial ótima, minimizando as perdas de potência ativa totais. O escopo do trabalho foi resolver o problema de reconfiguração ótima de redes de distribuição de energia elétrica aplicando uma metodologia inspirada no comportamento de colônia de formigas.

Para melhorar o desempenho, o algoritmo desenvolvido incorpora uma técnica de elitismo no processo de convergência mediante regras de atualização local e global do feromônio. Dos resultados optidos, conclui-se que o ACO proposto é promissor para a solução do problema de reconfiguração de redes apresentando um bom desempenho. Pretende-se agora investigar melhorias nas propriedades de convergência do algoritmo.

7. AGRADECIMENTOS

Os autores deste artigo agradecem á Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - CAPES e a Fundação de Amparo á Pesquisa do Estado da Bahia - FAPESB.

REFERÊNCIAS

Abdelaziz, A. Y., Osama, R. A., and El-khodary, S. M. (2012). Reconfiguration of distribution system for loss reduction using the hyper-cube ant colony optimization algo- rithm. IET Generation, Transmition and Distribution, 6(2):176–187.

Chiang, H.-D. and Jean-Jumeau, R. (1990). Optimal network reconfigurations in distribution systems:part 2 : Solution algorithms and numerical results. IEEE Transactions on Power Delivery, 5(3):1568–1574.

Ching-Tzong, S., Chung-Fu, C., and Ji-Pyng, C. (2005). Distribution network reconfiguration for loss reduction by ant colony search algorithm. Electric Power Systems Research, 7:190–199.

Ding, F. and Loparo, K. A. (2012). A simple heuristic method for smart distribution system reconfiguration. Energytech, IEEE Conference, pages 1–6.

Dorigo, M. and Stutzle, T. (2004). Ant colony optimization. MIT Press, 3(2):12.

Haque, M. H. (2000). A general load flow method for distribution systems. Electric Power System Research, 54(1):47–54.

Jeon, Y., Kim, J. C., Kim, J., Shin, J. R., and Lee, K. (2002). An efficient simulated annealing algorithm for network reconfiguration in large-scale distribution systems. IEEE Transactions on Power Delivery, 17:1070–1078.

Lin, W. M., Cheng, F. S., and Tsay, M. T. (2000). Distribution feeder reconfiguration with refined genetic algorithm. IEE Procedings Generation, Transmition and Distribution, 147(6):349–354.

Lu, L., Liu, J., and Wang, J. (2009). A distributed hierachical structure optimization algorithm based poly-partical swarm for reconfiguration of distribution network. SU- PERGEN 09 International Conference on Sustainable Power Generation and Supply, pages 1–5.

Olamaei, J., G., G., and T., N. (2007). An approach based on particle swarm optimization for distribution feeder reconfiguration considering distributed generators. Power Systems Conference: Advanced Metering, Protection, Control, Communication, and Distributed Resources, pages 326–330.

Parada, V., Ferland, J. A., Arias, M., and Daniels, K. (2004). Optimization of electrical distribution feeders using simulated annealing. IEEE Transactions on Power Delivery, 19:1135–1141.

Capítulo 2

10.37423/220606192

APLICAÇÃO DE UM SISTEMA WEBGIS COMO FONTE CONFIÁVEL DE CONSULTA DE DADOS EPIDEMIOLÓGICOS

Danilo Oliveira dos Reis Nascimento — *Universidade Federal do Estado de Mato Grosso*

Allan Biagio Hermann — *Universidade Federal do Estado de Mato Grosso*

Mateus Reis Santos — *Universidade Federal do Estado de Mato Grosso*

Paulo Henrique Cassiano Machado — *Universidade Federal do Estado de Mato Grosso*

Matheus Matos de Souza — *Universidade Federal do Estado de Mato Grosso*

Gracyeli Santos Souza Guarienti — *Universidade Federal do Estado de Mato Grosso*

RAONI FLORENTINO DA SILVA TEIXEIRA — *Universidade Federal do Estado de Mato Grosso*

JOYCE ALINE DE OLIVEIRA MARINS — *Universidade Federal do Estado de Mato Grosso*

Resumo. Devido a pandemia de 2020, muitas pessoas deixaram de dar a devida atenção a outras enfermidades e, por causa do isolamento prolongado, houve também um aumento da desinformação acerca de assuntos que não fossem correlacionados a Covid-19 ou ao isolamento. Portanto, com um aumento de disseminação de informações pelos meios de comunicação, qualquer informação publicada nestes meios poderia ser levada em consideração pela população, visto que este meio permite o compartilhamento de notícias falsas, ou também chamadas de *fake news*. Diante deste cenário, este artigo propôs o desenvolvimento e a implementação de uma aplicação *web* capaz de ser uma fonte viável de dados e informação de forma a permitir a um usuário comum acessar informações de certas enfermidades epidêmicas da região do estado de Mato Grosso.

Palavras-chave. Webgis; Fake News; Datasus; Dados Geoespaciais.

1.INTRODUÇÃO

Em 2020, a população mundial lidou com um evento atípico: uma pandemia mundial. Neste ano, pessoas ficaram em isolamento tendo como única fonte de informação a internet, levando as pessoas a direcionarem sua atenção predominantemente ao vírus, abrindo espaço para a propagação de desinformação e a falta de atenção a outras doenças (SILVA et al., 2020).

O conceito de Fake News, ou notícias falsas em uma tradução direta, já era abordado muito antes do início da pandemia, pois, com a adesão em massa das pessoas para as mídias sociais, a população acabou sendo exposta a informações criadas por terceiros, consequentemente, um mercado foi surgindo, sendo este dependente de números de visualizações, mesmo que para isso seja necessário criar notícia sensacionalista ou até mesmo falsa para gerar engajamento nas postagens (DELMAZO; VALENTE, 2018). Portanto, com uma tendência cada vez maior de fake news nas redes sociais e junto ao fato deste meio ter se tornando um grande meio de comunicação durante o isolamento de 2020, a população esteve à mercê do que esses meio de comunicação publicavam, não havendo nenhuma fonte de informação diversa que disponibilizasse dados de uma maneira simples, direta e confiável.

Desta forma, este trabalho teve como objetivo desenvolver e propor uma aplicação web que permita um usuário utilizar ferramentas simples, como busca de uma região ou um compilado de dados de um mapa digitalizado, com o foco em doenças, como Dengue, Zika e Chikungunya visto que estas doenças tiveram uma atenção reduzida durante a pandemia. Assim, o intuito foi fornecer uma base de dados sólida e expansível, além de disponibilizar parte dos dados de maneira simplificada, atendendo tanto um público acadêmico, que queira acessar uma base de dados, quanto parte da sociedade que queria apenas se informar de conscientizar a população da magnitude de outras comorbidades na região de Mato Grosso.

Foi desenvolvida uma aplicação webgis que disponibiliza dados de doenças que tiveram um aumento ou permaneceram presentes ao longo da pandemia e estes dados então são apresentados em forma de mapas com informações epidemiológicas de cada região dentro do estado de Mato Grosso, fornecendo uma fonte confiável de informação.

A aplicação web foi proposta considerando seu acesso fácil e intuitivo, sendo feita por meio de um navegador, além de que ao trazer uma visualização de dados através de mapa, ocorre uma interação direta do usuário com a informação de forma objetiva. É importante ressaltar que por ser um projeto

inicial, foi proposto apenas alguns dados epidemiológicos, entretanto, essa base de dados pode ser facilmente expandida, podendo armazenar e disponibilizar outros dados.

Por fim, essa aplicação pode ser eficaz para se tornar uma fonte de dados confiável, se distanciando do conceito de fake news e de desinformação visto que todos os dados disponibilizados serão checados e provenientes de uma fonte de dados relevante.

2. FUNDAMENTAÇÃO TEÓRICA

Um dos principais meios de compartilhamento de informação é através de páginas da internet, seja por meios de redes sociais, sites ou blogs. É importante observar que a relação da sociedade com a internet e, sobretudo, com as mídias sociais e sites de notícias, tem transformado a visão da população, com a adição desse meio em suas rotinas e estilo de vida (TARDIVO; VEDELLI, 2012).

Um dos assuntos abordados por Magrani (2014), em seu livro “Democracia Conectada”, é o conceito filter bubble (ou fitros-bolhas) que pode ser definida “como um conjunto de dados gerado por todos os mecanismos algorítmicos utilizados para se fazer uma edição invisível voltada à customização da navegação on-line” (MAGRANI, 2014). A partir desse conceito é importante mostrar que com a crescente mudança na rotina das pessoas, adicionando a internet como principal meio de obtenção de informação e interação, as pessoas se tornaram acomodadas diante à agilidade de acesso de informação, enquanto, o conteúdo é previamente filtrado para atender os gostos particulares de cada pessoa.

Ademais, como abordado no exemplo de seu livro, Magrani (2014), expõe uma prática feita pela Netflix, de fornecer um acervo de filmes aos usuários, porém, fazendo uma triagem prévia do que o usuário pode assistir, não barrando o acesso a outros filmes fora das sugestões além de fornecer um outro produto em que a pessoa possa ver sem gastar muito tempo procurando.

Nesse contexto é possível fazer uma comparação com o que ocorreu durante a pandemia de 2020, com o começo do isolamento social houve uma mudança de rotina das pessoas, tendo um aumento no uso de internet (HERNÁNDEZ; MEDINA, 2021). Logo, o principal assunto divulgado nas redes era sobre Covid-19, e, conforme essa demanda aumentava, mais assuntos e sites acerca desse tópico eram recomendados para os usuários, criando um ambiente de incertezas visto que o filtro abordado por Magrani (2014) não distinguia se uma notícia era verdadeira ou não, apenas se ela era relevante ao usuário.

Em vista desse espaço um novo conceito foi levado em conta durante o isolamento, as Fake News (ou notícias falsas), as quais consistem em notícias falsas disseminadas pela internet. Como abordado anteriormente, com o isolamento as pessoas aderiram em suas rotinas um maior uso da internet, seja para acessar notícias ou mídias sociais, entretanto, com esse aumento também trouxe uma maior demanda por informação, principalmente por informações da doença do coronavírus. Em meio dessa demanda, houve a abertura para disseminação de Fake News sobre o covid-19, ou seja, conforme o enfrentamento da doença aumentava e medidas eram tomadas mais notícias falsas eram espalhadas, pois não havia uma informação definitiva acerca do assunto conforme o aumento da demanda por notícias (NETO et al., 2020).

Devido ao aumento das Fake News, surge um espaço para a criação de um meio que pudesse combate-las. Assim, um dos meios abordados por Spinelli e Santos (2018) para o combate de Fake News consiste em buscar fontes confiáveis de informação e buscar veículos de comunicação com níveis altos de credibilidade (SPINELLI; SANTOS, 2018). Porém, para validar as informações de uma notícia, a pessoa precisa buscar as fontes dela e verificar se vem de uma fonte confiável. Entretanto, este processo pode acabar sendo exaustivo ou simplesmente inviável, já que a demanda por informação ágil também é um tópico a ser levado em consideração. Por conseguinte, uma alternativa para combater a Fake News, seria de fornecer uma fonte confiável de dados, além de permitir ao usuário uma experiência breve para acessar e assimilar o que está sendo transmitido.

Devido ao fato de a doença do coronavírus foi um dos principais alvos das notícias falsas durante a pandemia e o isolamento, esse trabalho propõe abordar um assunto na área de dados epidemiológicos, utilizando uma das fontes de dados confiável fornecida pelo DATASUS, através da ferramenta TABNET.

Conforme abordado por Correia (2009), um dos meios mais utilizados por mídias é o uso de infográficos, pois trazem as informações de maneira breve e clara. Ademais, esse meio de exposição de informação se mostra bem efetivo para o leitor assimilar e armazenar as informações contidas no infográfico (CORREIA, 2009). Todavia, um infográfico consiste justamente de uma imagem com informações, o que pode levar um tempo para assimilar e absorver o conteúdo, logo uma forma de interação usuário-informação é necessária.

Assim, uma maneira de poder fornecer informação de uma região mapeada, é através de um sistema de informação Geográfica (SIG), esse tipo de ferramenta dinâmica, chamada de WebGIS, permite a disponibilização de dados de uma região além de oferecer uma interação breve e fluída com os dados

contidos no mapa (SILVA et al., 2013). Com a tecnologia WebGIS, a qual permite colocar uma aplicação geográfica no browser, é possível conectar os dados coletados de um banco de dados e fornecer uma informação, a qual não está formatada, em uma aplicação voltada para dados epidemiológicos, sendo capaz de fornecer dados de uma região inteira com a quantidade de doentes, quantos infectados houveram no ano ou a variação de infectados durante um período de tempo.

3. METODOLOGIA

Para a criação do projeto, foram necessárias três etapas principais: divisão em programas para o servidor, definição/estruturação de linguagens de programação e os meios para manipular e obter os dados.

De forma a solucionar os problemas de desinformação e fake news relatados anteriormente, este estudo foi dividido em duas pesquisas: buscar uma fonte confiável de dados e um meio de apresentar os dados para a população.

A fonte utilizada como base de dados é o aplicativo Tabnet, o qual disponibiliza os dados coletados pelo Sistema Único de Saúde[1] . Enquanto que a parte responsável para disponibilizar os dados consiste em um conjunto de tecnologias que permite a construção de uma aplicação Webgis e, desta forma, irá permitir disponibilizar ao usuário o acesso a cópia dos dados coletados do Tabnet, porém, com uma interface mais amigável, intuitiva e interativa.

3.1. CRIAÇÃO DO SERVIDOR

Para a criação da parte do servidor da aplicação web é necessário dividir o servidor em duas partes, a parte do servidor web e a do servidor local.

Com o intuito de disponibilizar uma aplicação na web dinâmica, é necessário um servidor web capaz de lidar tanto com as requisições HTTP, que consistem em requisições de chamada, quanto da programação feita para o website. As requisições HTTP básicas, de chamar a página estática, não são capazes de fornecer a aplicação dinâmica idealizada no trabalho, para isso é necessário a tecnologia servlet, presente no apache TomCat. Sendo assim, o TomCat é configurado como o intermediário de fornecer a página e também para executar o programa, e a partir desta execução o servidor web é capaz de acessar o servidor local da máquina, local que será armazenado os dados utilizados no WebGis.

Com o servidor web configurado é necessário programar um servidor local. Esse servidor será responsável por armazenar os dados que serão contidos na aplicação. Para armazenar esses dados é então feito um sistema de gerenciamento de banco de dados. Sendo escolhido para essa tarefa o PostgreSQL. Esse sistema é um Sistema de Gerenciamento de Banco de Dados (SGBD) objeto-relacional, sem relação com linguagens de programação orientadas a objetos, isso permite adicionar novos tipos de dados, funções e operadores. Como a aplicação consiste em uma aplicação WebGis, ou seja, trabalha com dados vetoriais, é necessário um banco de dados capaz de lidar com dados vetoriais. Devido a possibilidade de expansão e modificação do PostgreSQL, foi instalado em conjunto ao SGBD, a extensão Postgis, responsável por introduzir a possibilidade de armazenar e interagir com dados vetoriais.

3.2. CRIAÇÃO DA APLICAÇÃO

Para que a aplicação seja criada, é necessário produzir um código que irá compilar e produzir um sistema para o usuário interagir. Neste trabalho foram utilizadas três linguagens para a produção de toda a aplicação, as quais foram: Html, CSS e Javascript.

HTML, Linguagem de marcação de hipertexto, é a junção de vários elementos, ao quais são conectados e estabelecem um conjunto de dados, armazenamento e compartilhamento de informações. Consiste na base da aplicação web, permite o desenvolvimento de websites e a inserção de novas funcionalidades, como vídeos e figuras, por meio de hipertextos.

Para que seja feito a estilização da página é necessário utilizar do CSS (Cascade Style Sheets), essa parte da programação consiste em inserir animações, sombreamento e estilos diferentes para botões e formulários. Conforme visto na imagem X, essa etapa do projeto delimita onde serão colocados as imagens e algumas funcionalidades, porém essas funcionalidades serão adicionadas através da programação em Javascript.

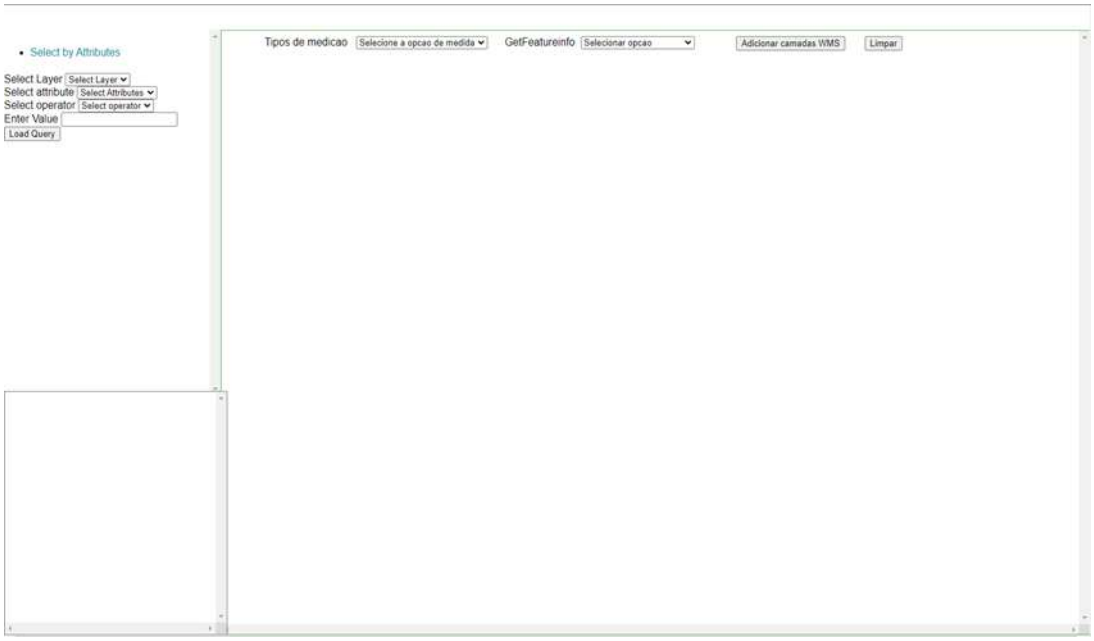


Figura 1 Interface da aplicação sem Javascript

O Javascript funciona como uma extensão para o código HTML, sendo uma linguagem que permite a programação orientada a objetos. Responsável por permitir uma aplicação adicionar comportamentos interativos às páginas da web. Ademais esta linguagem de programação permite adicionar bibliotecas capazes de fornecer funções e scripts que possibilitam trabalhar com os dados vetoriais armazenados no banco. Uma das bibliotecas utilizadas para a programação da aplicação web foi a Openlayers. Esta biblioteca Javascript é responsável por exibir dados de mapas em navegadores, dados vetoriais e marcadores de qualquer fonte de dados, o que permite criar aplicativos geográficos baseados em web, conforme visto na figura 2 que apresenta a interface do sistema WebGis rodando com todas as funcionalidades programadas.

3.3. FORMATAÇÃO DOS DADOS

Como forma de delimitar os dados que serão usados para criação da aplicação, foi utilizado apenas os dados das doenças Dengue, Febre Zika e Chikungunya no estado de Mato Grosso.

Após estabelecer toda a estrutura da aplicação é necessário inserir os dados que seriam expostos para os usuários, e para que seja possível disponibilizar os dados das doenças dengue, febre zika e Chikungunya, é necessário fazer dois procedimentos. O primeiro consiste em acessar o tabnet para a coleta de dados de pessoas infectadas, enquanto, o segundo procedimento para disponibilizar os dados em um mapa na aplicação, é necessário utilizar uma shapefile do estado de Mato Grosso para inserção de dados. Ressalta-se que toda a edição de dados é feita no software Qgis.

4. RESULTADOS

O resultado obtido durante a convecção do projeto foi a aplicação Webgis funcional. A interface do Webgis consiste em uma página web que permite a checagem de dados epidemiológicos no estado.

A interface do webgis apresentada na Figura 1, tem como foco primário chamar a atenção para o mapa, por isso boa parte dos outros elementos estão em cores menos chamativas. Um dos elementos que pode ser elencado é o menu de sobreposições, localizado no canto superior direito da Figura 1, este sendo responsável por permitir ao usuário qual dos dados será visto, vale ressaltar que por padrão todos os dados já vêm selecionados. No lado esquerdo superior tem-se a caixa de seleção de dados, esta parte da interface é responsável por selecionar os atributos da shape selecionada e disponibilizá-las para o usuário. Logo abaixo da caixa de seleção de dados, está a área de legendas que disponibiliza ao usuário a cor de cada shape.

Como o mapa é totalmente interativo, sendo possível visualizar todo o mapa mundi, é disponibilizado ao usuário alguns botões de interação, como zoom in e zoom out, um botão para centralizar novamente no mapa do estado de mato grosso e caso haja necessidade um botão para deixa a aplicação em tela cheia. Pelo fato de o mapa ter um visual chamativo é possível ao usuário clicar em qualquer parte das camadas de Casos para ter acesso a informações simplificada, sendo elas o nome e a quantidade de pessoas com essa doença nesse município.

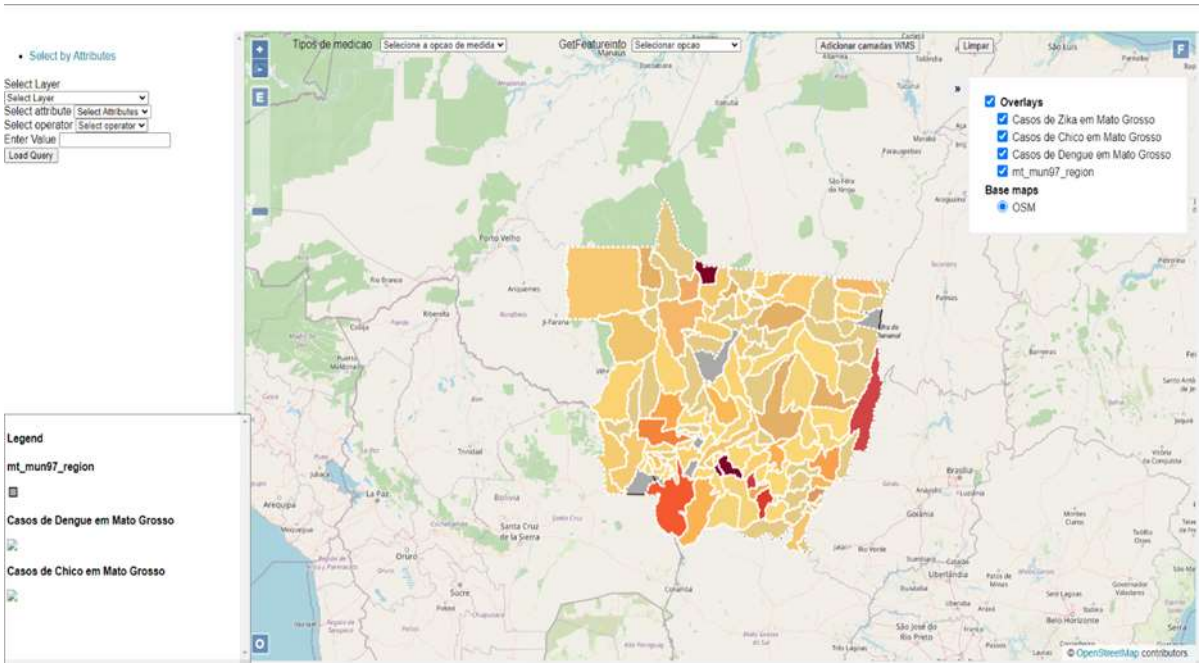


Figura 2 Interface da aplicação

Por padrão, o programa disponibiliza quatro shapes pré-definidas no painel de shapes:

• mt_mun97_region

• Casos de zika

• Casos de Chico

• Casos de Dengue

Estas shapes foram organizadas de tal forma a fornecer níveis de camadas as shapes, conforme algumas shapes são selecionadas elas serão sobrepostas. Essa possibilidade de seleção é vista na Figura 2. Caso seja necessário visualizar uma camada específica é necessário selecionar apenas a camada requerida, também é possível do usuário adicionar camadas extras conforme queira, como na Figura 2 que mostra a feição mt_zika_ que está presente somente no banco de dados do geoserver.

Figura 3 Lista de shapes

Após ter selecionado qualquer shape no mapa, o usuário pode acessar qualquer dado do mapa e realizar uma pesquisa específica a uma área do mapa, a qual esteja selecionada como uma das camadas dentro do mapa global. Para isso é necessário ativar a função *GetFeatureInfo* e, para ocorrer essa ativação, o usuário necessita selecionar uma das opções dentro da caixa de seleção conforme apresentada na Figura 3, sendo disponibilizado duas opções Ativar *GetFeatureInfo* e Desativar *GetFeatureInfo*. Após selecionar a opção Ativar *GetFeatureInfo*, enquanto ela estiver habilitada, qualquer clique feito em um dos municípios irá gerar um bloco com todas as informações contidas no polígono, como pode ser visto na Figura 4. Enquanto, a opção Desativar *GetFeatureInfo* é responsável por desabilitar a função, caso tenha um bloco ativo ele será fechado, e permitir ao usuário clicar em qualquer parte sem ativar o bloco de informação.

Figura 4 Caixa de seleção GetFeatureInfo e Botão para adição de camada

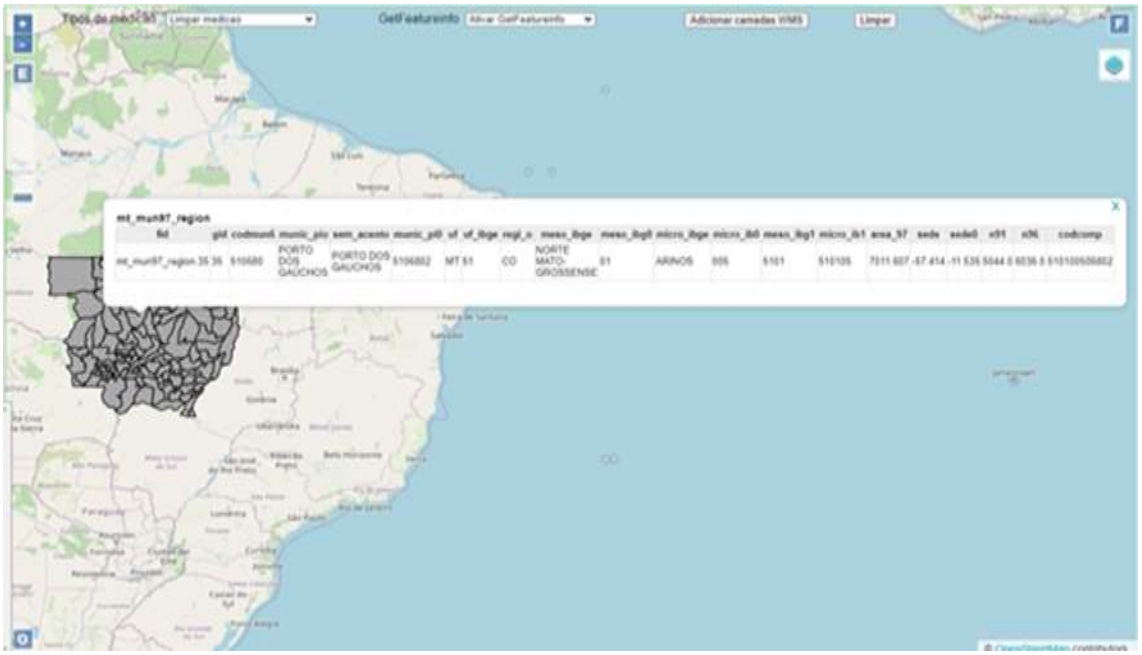


Figura 5 Resultado da Busca pelo GetFeatureInfo

Caso seja necessário fazer uma pesquisa específica, é possível ao usuário pesquisar dentro de uma layer os dados que ela busca.

Como pode ser visto na Figura 5, foi feita a busca por Cuiabá na shape do estado de Mato Grosso. Para isso foi inserido no campo Select Layer a layer que está presente no Geoserver, mt_mun97_region, ao selecionar qual shape utilizar no campo Select Atribute é disponibilizado ao usuário os atributos presentes na tabela.

No exemplo, foi escolhido o campo que possui os nomes dos municípios sem acento, após selecionar o atributo o usuário precisa escolher o operador para a busca, para pesquisas que o campo possua caracteres será disponibilizado a opção Like, feita a escolha do operador o usuário fornece o que será buscado no banco, nesse exemplo foi o nome da cidade Cuiabá, após todos os parâmetros fornecidos o usuário aperta o botão Load Query e após alguns instantes ele será direcionado para a shape com as informações do campo encontrado.

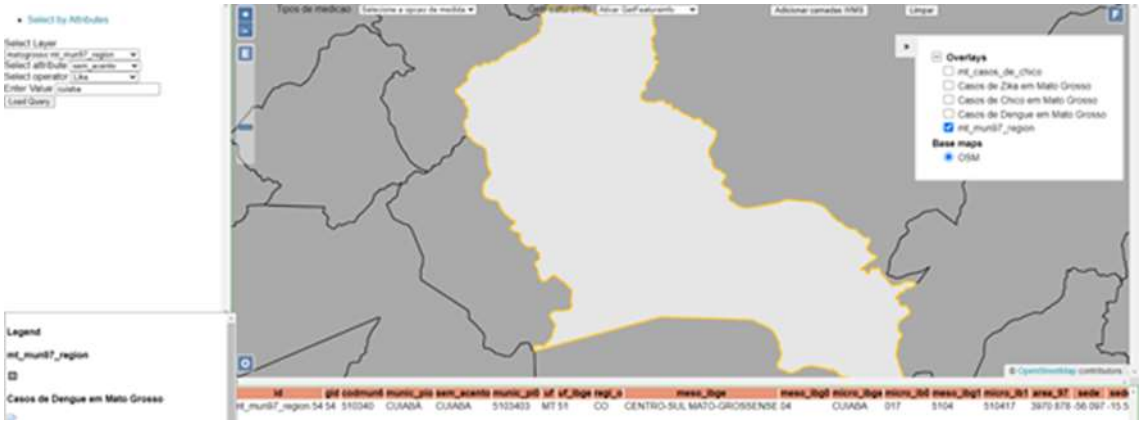


Figura 6 Busca por Cuiabá na shape

4. CONCLUSÃO

Neste trabalho foi produzido um sistema webgis para acompanhamento de dados epidemiológicos, preparado para armazenar e apresentar os dados pegos do banco de dados do DATASUS. A aplicação permite ao usuário observar e analisar o avanço de algumas doenças em cada município do estado de Mato Grosso. Além dos dados disponibilizados no trabalho, é possível adicionar dados de outras doenças, tanto do estado de Mato Grosso quanto de qualquer outro estado. Para isso basta adicionar os dados no banco de dados sendo utilizado na aplicação.

Para mais, a abordagem deste trabalho trouxe uma visão mais ampla de como uma aplicação WebGis pode trazer os dados a um usuário, reforçando o uso de algumas tecnologias atuais capazes de oferecer uma interação usuário-aplicação mais satisfatória. Ademais em trabalhos similares, que abordam o uso da ferramenta webgis, como no trabalho produzido por Silva et al. (2013) que traz a discussão sobre a criação de um sistema WebGis para promover a atividade turística, ambos os trabalhos trazem o uso de mapas como forma de trazer a informação, mas diferem na maneira que ele é utilizado e os dados que são fornecidos. Devido a diferença dos dados utilizados, sendo neste trabalho os dados epidemiológicos em cada município de Mato Grosso e o trabalho produzido por Silva et al.(2013) trás uma abordagem com imagens e texto de cada parte turística da região abordada. Desta maneira ambos os trabalhos trazem um uso diferente da ferramenta webgis para trazer atenção e foco a um determinado tema ou informação.

Outrossim, uma das formas de validar a eficiência dessa aplicação será através da coleta dos dados de tráfego de pessoas ao longo de um tempo, além do tempo médio de cada usuário no site. Essa forma de validação é uma das maneiras de verificar se a aplicação está sendo eficiente em espalhar informação e de prender um usuário por tempo suficiente para analisar o que está sendo apresentado.

Por fim, pela qualidade dos dados coletados pelo DATASUS, há uma credibilidade em cima das informações fornecidas pela aplicação o que permite uma nova visão para os dados coletados ao longo dos anos pelo DATASUS e fornece uma nova fonte de dados para as pessoas utilizarem esses tipos de dados, seja para pesquisas ou para o aumento de conscientização destas doenças no estado.

5. REFERÊNCIAS

DELMAZO, C.; VALENTE, J. C. L. Fake news nas redes sociais online: propagação e reações à desinformação em busca de cliques. Media & Jornalismo, [S. l.], v. 18, n. 32, p. 155-169, 2018. DOI: 10.14195/2183-5462_32_11. Disponível em: https://impactum-journals.uc.pt/mj/article/view/2183-5462_32_11. Acesso em: 7 mai. 2022.

SILVA, C. M. D. et al. A Pandemia de COVID-19: Vivendo no Antropoceno. Revista Virtual de Química, Niterói-RJ, v. 12, n. 4, p. 1001-1016, jun./2020.

CORREIA, J. S.; Concepção e Implementação de um WebSIG no Parque Nacional da Gorongosa usando software de cáodigo aberto e livre. Instituto Superior de Estatística e Gestão de Informação da Universidade Nova de Lisboa, 2011. Disponível em: Acesso: 10 fevereiro 2022.

MOTA, H. H. C. S; WebSIG Concepção e Desenvolvimento. Universidade do Porto, Faculdade de Letras, Portugal, 2013. Disponível em: Acesso em: 05 fevereiro 2022.

INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS – INPE Fundamentos de Geoprocessamento – Tutorial. DPI-INPE, 2002

SPINELLI, E. M.; SANTOS, J. DE A. JORNALISMO NA ERA DA PÓS-VERDADE: fact-checking como ferramenta de combate às fake news. Revista Observatório, v. 4, n. 3, p. 759-782, 29 abr. 2018.

CORREIA, Marcos Balster Fiore. A Comunicação de Dados Estatísticos por intermédio de Infográficos: Uma Abordagem Ergonômica. Anamaria de Moraes. 472 f. Programa de Pós-Graduação em Artes, PUC-Rio, Rio, 2009.

SILVA, Norberto Peçanha da. A utilização dos programas TABWIN e TABNET como ferramentas de apoio a disseminação das informações em saúde. 98 f. Dissertação (Mestrado Profissional em Saúde Pública) - Escola Nacional de Saúde Pública Sergio Arouca, Fundação Oswaldo Cruz, Rio de Janeiro, 2009.

SILVA, K. G. D. et al. Elaboração de mapa interativo em Webgis como meio de promover a atividade turística: um experimento na rota sede - Nova Lima (MG). Territorium Terram, Minas Gerais, v. 1, n. 2, p. 107-122, mar./2013. Disponível em:

http://www.seer.ufsj.edu.br/index.php/territorium_terram/article/view/325. Acesso em: 29 abr. 2022.

TARDIVO, Jessica; BIZELLI, José. Cibercultura: a internet como meio de comunicação e sociabilidade contemporânea. Disponivel em:

<https://www.researchgate.net/profile/Jose-Bizelli/publication/279718646_Cibercultura_a_internet_como_meio_de_comunicacao_e_sociabilidade_contemporanea/links/5598899508ae793d137e1b41/Cibercultura-a-internet-como-meio-de-comunicacao-e-sociabilidade-contemporanea.pdf> Acesso em 22 de maio de 2022

HERNÁNDEZ, Rubén Cervantes; MEDINA, P. M. C. Transformations in communication and sociability habits through the increased use of socio-digital networks in times of pandemic. Ámbitos Revista

Internacional de Comunicación, Universidade de Sevilla, Espanha, v. 1, n. 52, p. 37-51, fev./2021. Disponível em: https://idus.us.es/handle/11441/107676. Acesso em: 10 mai. 2022.

NETO M, GOMES T de O, PORTO FR, RAFAEL R de MR, FONSECA MHS, NASCIMENTO J. Fake news no cenário da pandemia de Covid-19. Cogitare enferm. [Internet]. 2020 [acesso em “08 de maio de 2022”]; 25. Disponível em: http://dx.doi.org/10.5380/ce.v25i0.72627.

Capítulo 3

10.37423/220606194

HIGH RESOLUTION FINGERPRINT SYNTHESIS

Matheus Matos de Souza — *Universidade Federal de Mato Grosso*

Allan Biagio Hermann — *Universidade Federal de Mato Grosso*

Danilo Oliveira dos Reis Nascimento — *Universidade Federal de Mato Grosso*

Mateus Reis Santos — *Universidade Federal de Mato Grosso*

Paulo Henrique Cassiano Machado — *Universidade Federal de Mato Grosso*

Gracyeli Santos Souza Guarienti — *Universidade Federal de Mato Grosso*

RAONI FLORENTINO DA SILVA TEIXEIRA — *Universidade Federal de Mato Grosso*

JOYCE ALINE DE OLIVEIRA MARINS — *Universidade Federal de Mato Grosso*

Abstract: *In the last few years, laws that restrict the use of biometric data have been implemented in many countries, for example, the General Data Protection Act in Brazil. One effect of this restriction is currently all high resolution fin- gerprint databases are no longer publicly available. To deal with this problem we developed in this work a generative algorithm for fingerprint pore synthesis. The experiments showed that our constructed algorithm using Variational Au- toencoder respects the spatial distribution of pores. As far as we know, this is the first work in the literature in this direction.*

1. INTRODUCTION

Fingerprint has been used to identify people and solve crimes since the 19th century, be- ing considered as evidence in trials [Barnes 2013]. This is due to the fact that the details of the papillary ridges are unique and remain virtually unchanged throughout life. Even identical human twins have distinct fingerprints [Fierro 2020]. Advances in acquisition technologies (from ink and paper, talc and chemical treatments to tiny sensors) allow people today to use the unique information present on the surface of their fingers as au- thentication in, for example, banking applications from their smartphones. Patterns in fingerprint can be divided into three levels. The first level, known as global, concerns the orientation of the papillary ridges. At this level, singular points known as delta and core define variations in the direction of the ridges. There also are other features at this level such as external fingerprint format, orientation and frequency. The level 2 (two) deals with local features, known as minutiae. The two most notable types of minutiae are terminations and bifurcations. The level 3 (three), known as very-fine level, is the most detailed level. At this level, features over the ridge can be seen. These details include width, shape, curvature, ridge contours, and other permanent details like incipient points and ridges [Maltoni et al. 2009]. It is also at this level that it is possible to find the pores of fingerprints. Pores are located on the papillary ridges and their use significantly increases the distinction of fingerprints. According to researchers, the use of pores in fingerprint recognition improves detection by 10 times, making recognition more accurate, lowering the risk of incorrect authentication or forgery [SATO 2018].

The advantage of using pores in identification is notable [SATO 2018]. How- ever, challenges still need to be overcome. A major challenge is related to the avail- ability of annotated data. In fact, the training of artificial neural networks and sev- eral other types of research involving high resolution fingerprints demand consider- ably large databases. Unfortunately, there is currently a shortage of such databases [Andre´ Brasil Vieira Wyzykowski and de Paula Lemes 2020]. Among the reasons for the scarcity, we can mention two: the first is that the extraction of samples of level 3 (three) of distinction must be done by specific equipment. In order to achieve the fine details of digital printing, high resolution (eg 1000 dpi) and good quality images are needed [Maltoni et al. 2009]. The second reason is the legal protection of biometric data. In recent decades, many countries have developed laws that restrict the use of biomet- ric data by their citizens, such as LGPD a Brazilian law that determines a more cautious treatment of sensitive personal data, which includes genetic or biometric data citepeixoto.

Some synthetic fingerprint generators such as Anguli are avail- able. However, these generators still do not reach level 3 (three) of dis- tinction (i.e. they do not yet generate high resolution fingerprint images) [Andre´ Brasil Vieira Wyzykowski and de Paula Lemes 2020]. It is precisely this gap that we are trying to fulfill. This study seeks to explore Generative Modeling focusing on the study of fingerprint pore generation. To achieve this goal, a network architecture known as Variational Autoencoder was implemented.

2. METODOLOGY

2.1. ARQUITETURA

In this project, we used a Variational Autoencoder presented in the official repository of the book Generative Deep Learning [Foster 2019]. Some modifications were made to deal with the particularities of the database. The Encoder consists of 4 convolutional layers that reduce the size of the inputs. After each convolutional layer, there are activation layers, in the case of this architecture, Leaky ReLU layers were used. In the normal ReLU, the output is zero for $x < 0$ and remains the same for values greater than or equal to x. As for Leaky ReLU there is α, a very small value that is often defined as $\alpha = 0.01$, where when x is a value less than 0, x is multiplied by α. By default, this value is set to $\alpha = 0.3$ in the Leaky ReLU layers of the API Keras [Versloot 2019].

The purpose of having that activation function is to increase nonlinearity in the database. The reason that is done is that images are naturally non-linear. The Leaky ReLU layer serves to break linearity even further in order to compensate for the linearity that can be imposed on an image after the convolution operation. Leaky ReLU is used instead of ReLU to avoid a problem known as “dying ReLU”, where too much data sparseness leads to the death of the network. The ReLU neurons that are stuck on the negative side of the network are dead and always produce 0, being unlikely to recover. These neurons play no role in discriminating the input and are essentially useless, over time this can take over much of the network to the point of that whole region also not performing any role [Liu 2017].

The flatten layer is responsible for removing the resulting dimensions from the last convolution layer and models the tensor to have a shape equal to the number of elements contained in it. The data that after the last convolutional layer had the format 32x32x64, happens to have a single dimension and 65536 elements after going through the flatten layer. After the flatten layer, a dense layer maps these elements to the mean vector (mu) and to the logarithm of variance vector (log-var). So, we have the output of Encoder. The latent space has 200 dimensions, unlike the original architecture which had a

latent space of two dimensions. The greater number of dimensions is due to the fact that this Autoencoder is not only intended to reconstruct handwritten digits (the purpose for which the book code was implemented), but rather fingerprint pores.

2.2. VORONOI DIAGRAM

It was necessary to use a good assesmenter to verify if what was coming out from the network were images in which the synthetic pores were arranged following the “natural order” that governs the arrangement of pores in real fingerprints. The concepts of the Voronoi diagram are fundamental to understand this assesmenter.

The implemented assessmenter takes into account the aspect of regularity, which states that fingerprint pores are arranged at regular intervals along the papillary ridges, and the biological evidence that leads to the spatial relationship existing in the natural distribution of pores. The optical configuration consists of points that coincide with the centroids of the respective Voronoi regions [Teixeira and Leite 2015]. The assessmenter in this work was implemented following the following steps: 1. The pore centroids of the generated images were calculated; 2. From these centroids the Voronoi regions were generated; 3. The centroids of the Voronoi regions were calculated; 4. Euclidean distances between the pore centroid and the Voronoi region centroid were calculated for each pore;

2.3. GENERATING PORES

The Variational Autoencoder receives the input data and maps it to two vectors, the mean vector and the variance vector. Those means and variances allow us to generate a multi- variate normal distribution from which a z point is sampled and passed to the Encoder to the reconstruction of that data. The RMSE and KL divergence cost functions respectively are responsible for ensuring that the network has the best weights and that it results in good reconstructions at the Decoder output and working so that the latent space of 200 dimensions is as organized as possible. Thas allows us to sample random values from the standard normal distribution that will be our z point and pass it through the Decoder, so we have a new generated pore image. Using that technique, pore images were generated, and those are distinct from any other image present in the database. The result can be seen in Figure 1, item (a).

The 150 threshold was applied to the pore images Figure 1 item (a) because this threshold value was the one that obtained the best results in the experiments, that is, with this value we obtained the best Euclidean distances with the best amount of reconstructed pores. Finally, we calculated the controids

of these new generated points that give us coordinates of the pores, as can be seen in Figure 1, item (b).

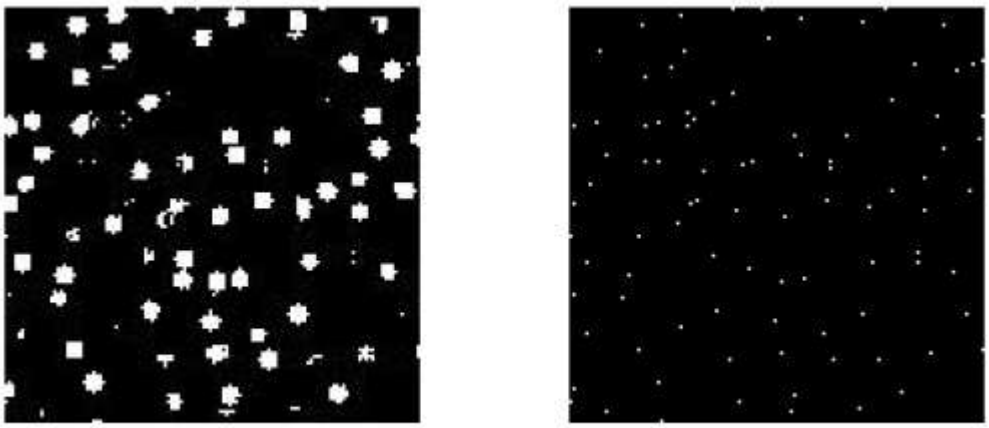


Figure 1. Generated pores. Item 0 corresponds to a output image and item (b) corresponds to its coordinates.

3. Experiments

3.1. Database

We used PolyUHR database which consists of 120 high quality fingerprint images with resolution of 480×640 pixels. The resolution of the images allows the visualization of pores in the fingerprint. To increase the amount of training and testing images, 128 × 128 pixel overlapping clippings were performed by applying horizontal windows to every 32 pixels of the image. With this processing, each sample from the original database became 204 samples to the new database. The new database has a total of 24480 samples, of which 18361 were used for training and 6119 were used for testing. The training was performed from images with points that represent the location of the pores of the high resolution fingerprint images. These representations were achieved due to manual annotation of pores of the images done by the project team.

3.2. Voronoi Diagram's assesmenter

This assesmenter was carried out basing on the concepts of Voronoi Diagram and the spatial distribution of pores, as seen above. The first step of the assessment is to calculate the centroids of all points of both reconstructed and original images. The centroid is the center of the pore representation of fingerprint images. Having the centroids of all pore representations in the image, the next step is to

calculate the Voronoi regions from all obtained centroids. But first, we need to convert the image from grayscale to black and white, in which pixels consist of just two colors. Considering that the grayscale images represented in two-dimensional arrays consist of pixels that have the possibility to assume 256 colors, we need to define a threshold for what will become white (1 in the array) and what will become black (0 in the array). This image change makes it possible to reduce noise, which are unwanted parts of the image that do not have a defined meaning. In this work, image pixels that do not contribute to the representation of pores are considered noise. But, on the other hand, it is possible with this noise reduction process eliminating parts of the image that collaborate to represent a pore. In view of this, three limit values were defined, which are 80, 150 and 200. The idea is that the Euclidean distance values between the centroids of the influence regions and the centroids of the pore regions of the reconstructed images are close to the values of Euclidean distance between the centroids of the original images.

3.3. Pore density

It is not enough just to know if the Euclidean distances between the pores and the centroids of their Voronoi regions are close, as it may happen that malformed pores are excluded in the transformation of the image from grayscale to binary. It is also necessary to know the number of pores formed in the reconstruction. A metric was implemented to be used for that. So a reconstructed sample is considered good when the Euclidean distances and the amount of pores are close to the ones in the original image.

The method used for the implementation of this metric is simple: the arrays of the reconstructed and original images were traversed with the threshold values 80, 150 and 200 to quantify the number of pores present on them and to have the pore density for the images with those threshold values. Then, the number of pores in each 16 x 16 pixel window was calculated. Thus, the pore density was obtained, which is the amount of pores for each 16 x 16 pixel window. After obtaining all densities for all images, the simple arithmetic mean was calculated for each experiment.

4. RESULTS

4.1. EXPERIMENT I

Figure 2 presents histograms with a threshold equal to 80. That is, all color values in the grayscale image array which are below 80 will be considered 0 (black) and all color values equal or greater than 80 will be considered 1 (white). The x-axis (horizontal axis) represents the Euclidean distance values between the pore centroid and the Voronoi region centroid. The y-axis (vertical axis) represents the number of occurrences. In orange we have the Euclidean distances for the reconstructed images and in lilac the Euclidean distances for the original images. Finally, it is possible to observe in this image the overlap of the two histograms. The histograms encompass values in a range from 0 to 50, the range in which most of the mean values of Euclidean distances of the samples used are found. To plot the histograms, the simple arithmetic mean of the Euclidean distances of each image was used.

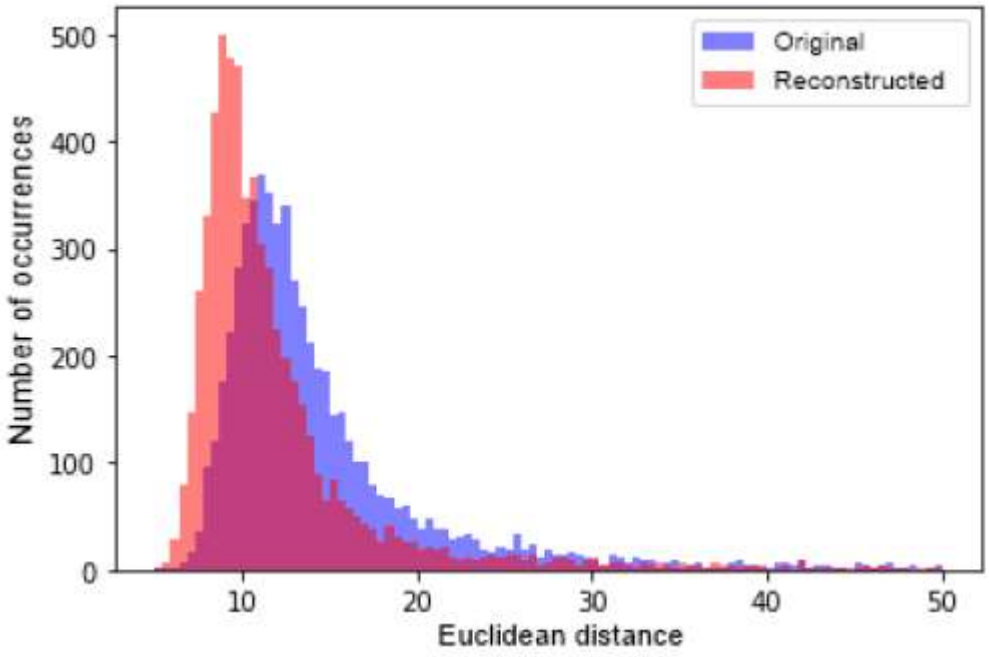


Figure 2. Number of occurrences for the Euclidean distances with threshold 80

For the reconstructed images using the threshold 80, we observed that the range from 0 to 30 concentrates most of the Euclidean distances. However, the part of the diagram with the largest number of occurrences is shifted further to the left in relation to the part with the largest number of occurrences of the original images diagram for this same threshold value. What can also be observed is that for the reconstructed images there is a greater number of occurrences in certain Euclidean distances, causing the diagram to reach a number close to 500 occurrences on the vertical axis.

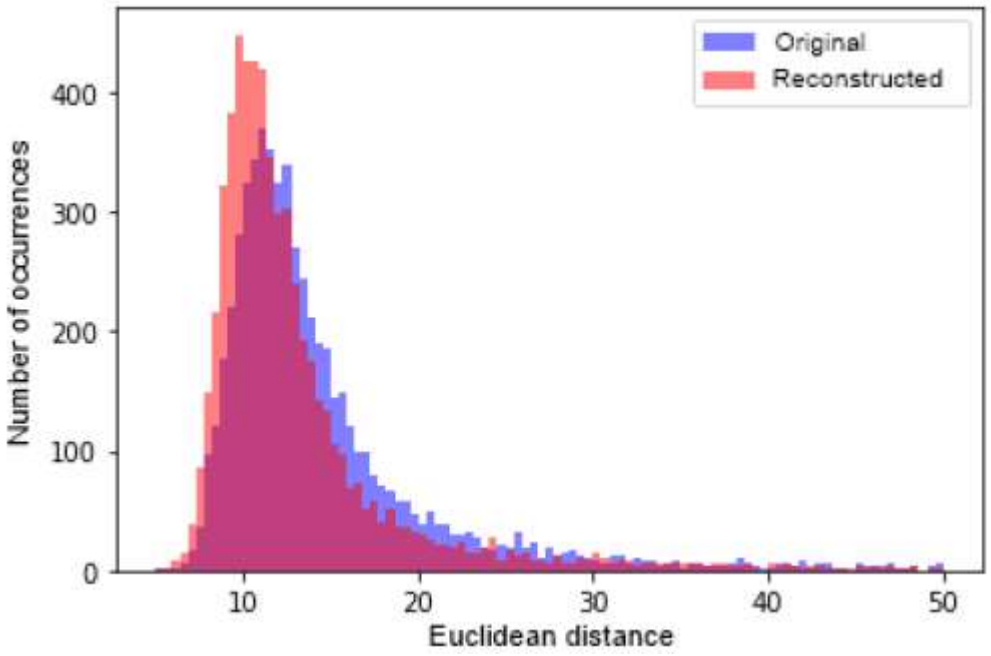


Figure 3. Number of occurrences for the Euclidean distances with threshold 150

The pink region represents the intersection of occurrences of Euclidean distances. The larger the intersection area, the better the images were reconstructed for the given threshold value, as the Euclidean distance values between the influence region centroid and the pore are similar.

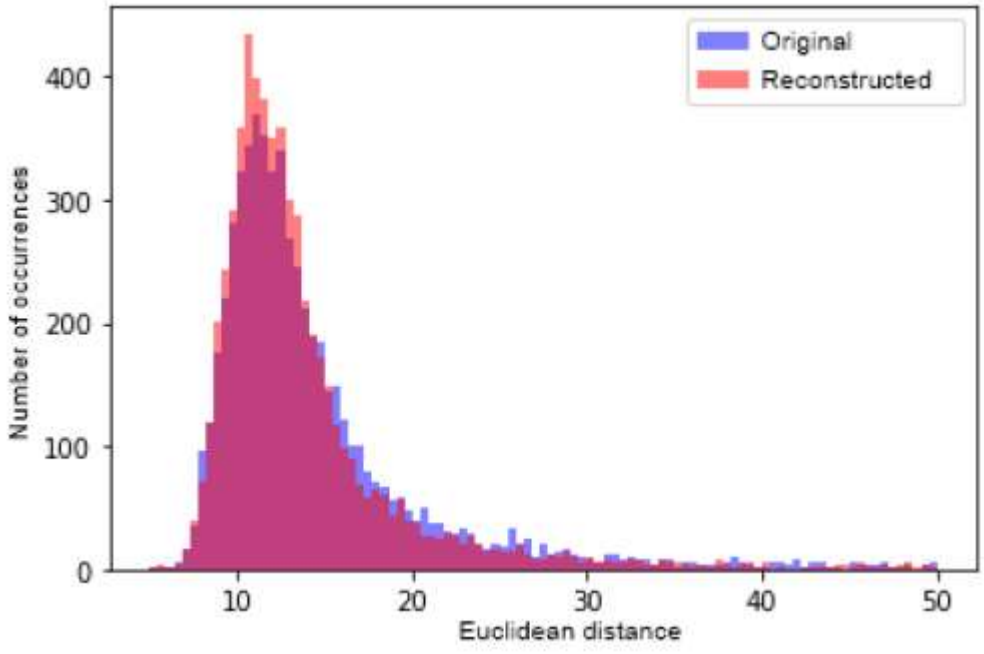


Figure 4. Number of occurrences for the Euclidean distances with threshold 200

As can be seen in the histograms for the threshold 150, in Figure 3, the graph in the region with the highest occurrence of reconstructed images is still slightly shifted to the left, whereas the graph of the

original images is slightly shifted to the right in relation to each other. The highest number of occurrences range still extends from 0 to 30.

In Figure 4, when threshold 200 is set, there is almost no detachment of the graph- ics, showing a better overlap between the original images and the reconstructed images. This represents similar Euclidean distances for images with this threshold value. The pore density analysis will be fundamental to conclude if this threshold value is what re- ally brings the best reconstructions.

4.2. EXPERIMENT II

In the analysis of the pore density mean, the threshold value that managed to have better reconstructions was the value 150, as it resulted in a density value closer to the density value for the original images, as can be seen in Table 1. The density value for the threshold

Table 1. Pore density mean for each threshold value

Threshold	Original	Reconstructed
80	1,156	1,511
150	1,156	1,153
200	1,156	0,952

80 shows that there were many noise pixels in the generate images that were wrongly considered in the pore representation. The density mean value for the threshold equal to 200 indicates that a significant amount of pixels that represented pores were disregarded, causing only the best reconstructed pores to be identified after the threshold. And the fact that only the best reconstructed pores are considered visually is why histograms to limit equal to 200 have a larger intersection area and look better, resulting in supposedly better reconstructions. .

5. DISCUSSION

As shown in [Teixeira and Leite 2015], the relationship between a given pore and its neighboring pores is not just linear over a ridge. Considering that, this work presents a new approach for synthesis of fingerprint pores, which is contrary to other works in the literature such as [Andre´ Brasil Vieira Wyzykowski and de Paula Lemes 2020] in which the pores are generated from the mean of the distances between consecutive pores in a pa- pilary ridge. Our study is a step towards a more realistic approach that takes into account how pores are spatially distributed.

The results were satisfactory, they showed that the set of obtained pores is dis- tributed with a pattern similar to the one of the database. However, some improvements are still needed. It was noticed that some pores were very close to each other. This is be- cause some pore representations were generated with malformations. When the threshold 150 was applied, due to the malformations, what would have been a single pore became 2 (two) or more very close ones. Erros like that meant that we did not have better results in the test database reconstruction experiments when obtaining the Euclidean distances between the pores and centroids of their Voronoi regions and when comparing the re- constructed images with the original images. Annotating pores in more high resolution fingerprint samples can be used to expand the database and, using a significantly larger database in training and testing, certainly better results would be obtained with better formed pores in those cases.

6. CONCLUSION

This work presented a generative algorithm for the synthesis of fingerprint pores. A dif- ferent approach was sought here from the approaches seen in other literatures, such as [Andre´ Brasil Vieira Wyzykowski and de Paula Lemes 2020] in which the pores are gen- erated from the mean of the distances between consecutive pores in a ridge of the fin- gerprint. But as seen in [Teixeira and Leite 2015] the spatial distribution of pores must be considered. There is a relationship between a given pore and its neighboring pores, and this relationship is not only linear with distances between the pores in the direction of their ridge. Thus, a more realistic approach should consider this relationship and the Generative Modeling with Variational Autoencoder works in that case, as it is understood that there is a natural rule for the generation of pores and through the Generative Mod- elling used in this study, a rule very similar to that rule was reached, with the generation of pores, considering the spatial relationship and the whole set of pores existing in certain fingerprints.

The results show that the set of pores synthesized by the algorithm follows a spatial distribution similar to that found in the database. The main contribution of this work is to demonstrate that it is possible to generate pores through an approach that does not infringe their spatial distribution, allowing that in the future there will be level 3 (three) of distinction fingerprint generators that will facilitate studies to improve security systems based on fingerprints through consistent databases. In future works, we will expand the database (manually annotating a larger set of images) and apply the methodology known as GAN (Generative Adversarial Network) for synthesis.

REFERENCES

Andre´ Brasil Vieira Wyzykowski, M. P. S. and de Paula Lemes, R. (2020). Level three synthetic fingerprint generation. arXiv, 1(1):1–12.

Barnes, J. G. (2013). The Fingerprint Sourcebook - History. U.S. Department of Justice - NIJ.

Fierro, P. P. (2020). Identical Twins and Fingerprints. Verywell Family.

Foster, D. (2019). Generative Deep Learning. O'Reilly Media, Sebastopol, CA, USA. Liu, D. (2017). A Practical Guide to ReLU. Medium.

Maltoni, D., Jain, A., Maio, D., and Prabhakar, S. (2009). Handbook of Fingerprint Recognition. Springer.

SATO, M. (2018). Sweat pores improve fingerprint recognition tenfold, researchers say.

Nikkei Asia.

Teixeira, R. F. S. and Leite, N. J. (2015). A new framework for quality assessment of high-resolution fingerprint images. IEEE, 1(1):1–24.

Versloot, C. (2019). Using Leaky ReLU with Keras. MACHINECURVE.

Capítulo 4

10.37423/220606219

COMPARATIVO DE TEMPO DE CRIPTOGRAFIA EM UM BANCO DE DADOS RELACIONAL

Gabriel Jaber Aguiar — *Faculdade de Engenharia – Universidade Federal do Mato Grosso (UFMT)*

Sergio Lucio Nunes da Silva — *Faculdade de Engenharia – Universidade Federal do Mato Grosso (UFMT)*

Lucas santos de Almeida — *Faculdade de Engenharia – Universidade Federal do Mato Grosso (UFMT)*

Jadson Matheus Bezerra de Lima — *Faculdade de Engenharia – Universidade Federal do Mato Grosso (UFMT)*

Gracyeli Santos Souza Guarienti — *Faculdade de Engenharia – Universidade Federal do Mato Grosso (UFMT)*

Joyce Aline de Oliveira Marins — *Faculdade de Engenharia – Universidade Federal do Mato Grosso (UFMT)*

Resumo. Com *o rápido crescimento do armazenamento de informações sigilosas, bancos de dados tornaram-se mais suscetíveis a ataques cibernéticos. Dessa forma a implementação de um método criptográfico visa dificultar a exposição de tais dados. Para tal, escolher qual método criptográfico será implementado depende de fatores como o tempo necessário para criptografar os dados. Portanto, este artigo busca fazer um comparativo de tempo entre a criptografia assimétrica RSA e simétrica AES e observar as diferenças de tempo ao encriptar uma coluna em um banco de dados relacional.*

1. INTRODUÇÃO

O avanço na popularização da Internet abriu muitas oportunidades de comunicação e transmissão de dados. Devido a este rápido crescimento a segurança existente entre os bancos de dados e as aplicações ficaram mais suscetíveis a ataques. Nesse contexto, um banco de dados armazena diversas informações havendo a possibilidade de conter dados sensíveis como RG, CPF, números de telefone, endereços, etc. Dessa forma torna-se essencial a implementação de um método criptográfico para evitar que as informações existentes sejam exibidas de forma explicita na rede caso tenha-se um vazamento.

Para alcançar este objetivo, fez-se necessário compreender os métodos de segurança criptográfica já estabelecidos. Dos diversos modelos existentes dois se des-tacam: as criptografias simétricas e assimétricas. Com modos de operações diferentes, existem variações de tempo e diferenças de segurança na implementação destes. Utili¬zando o sistema gerenciador de Banco de Dados Relacional (SGBD) Microsoft SQL Ser-ver, este artigo busca fazer a encriptação de dados de uma coluna em uma tabela e verificar a diferença do tempo entre as mais conhecidas criptografias simétricas e assimétricas res¬pectivamente: Advanced Encryption Standard (AES) e Rivest-Shamir-Adleman (RSA).

2.FUNDAMENTAÇÃO TEÓRICA

A palavra criptografia é derivada do grego Kryptós que em tradução literal significa oculto ou secreto. A criptografia procura maneiras de codificar uma mensagem de modo que somente o destinatário legitimo consiga interpreta-la. Desde os tempos antigos, foi muito utilizada ao longo da história [Coutinho 1999].

Algoritmos criptográficos tem como objetivo "esconder" informações sigilosas de qualquer pessoa desautorizada a lê-las, isto é, de qualquer pessoa que não conheça a chamada chave secreta de criptografia. A chave, na criptografia, é um valor fornecido como uma entrada para um criptossistema e funciona como um manual de como a men¬sagem deve ser criptografada contendo instruções do processo [Kim and Solomon 2014]. Algoritmos de encriptação por chave geralmente são divididos em duas categorias: de encriptação simétrica e de encriptação assimétrica. A grande diferença entre esses dois métodos se deve ao fato de que os algoritmos de cifragem simétrica utilizam de uma única chave que é usada tanto para encriptar quanto para decifrar o mesmo objeto, enquanto a encriptação assimétrica, que recebe também o nome de criptografia de chave pública, faz o uso de duas chaves

diferentes, uma é denominada chave pública e usada para criptogra¬far o objeto e a outra é chamada de chave privada e utilizada para decifrar este mesmo objeto.

2.1. CRIPTOGRAFIA AES

Criado a partir de uma competição internacional aberta desde 1997, o AES (Advanced Encryption Standard) foi inventado com o objetivo de escolher um substituto para o então método criptográfico simétrico mais utilizado até o momento, o DES. Coordenado pelo National Institute of Standards and Technology, NIST, dos EUA. Todos os candidatos deveriam atender aos seguintes requisitos [Terada 2008]:

- Bloco de 128 bits na entrada e na saída;
- Chave com tamanhos de 128, 192 ou 256 bits;
- Velocidade e segurança igual ou superior ao Triple-DES (uma forma mais avançada da criptografia DES);
- Tem que ser eficientemente implementável em softlhard/smart-card.

Em 2000 a criptografia de nome Rijndael foi selecionada como AES. A estrutura do algoritmo é feita da seguinte forma: A entrada para os algoritmos de encriptação e decriptação é um único bloco de 128 bits indicado como uma matriz quadrada de 4 x 4 bytes. Esse bloco é copiado para um array denominado Estado, que é modificado a cada etapa de encriptação ou decriptação e no final, é copiado para uma matriz de saída. A chave também é apresentada como uma matriz quadrada de bytes. Essa chave é expandida para um conjunto de palavras de chave onde cada palavra tem quatro bytes e o conjunto total de palavras para uma chave de 128 bits, por exemplo, é 44. A ordenação de bytes dentro da uma matriz de chaves é por coluna. Assim, os primeiros quatro bytes de entrada de texto claro de 128 bits para a cifra de encriptação ocupam a primeira coluna da matriz , os próximos ocupam a segunda coluna, e assim por diante. Da mesma forma, os primeiros quatro bytes da chave expandida, que forma uma palavra, ficam na primeira coluna da matriz w [Stallings 2014].

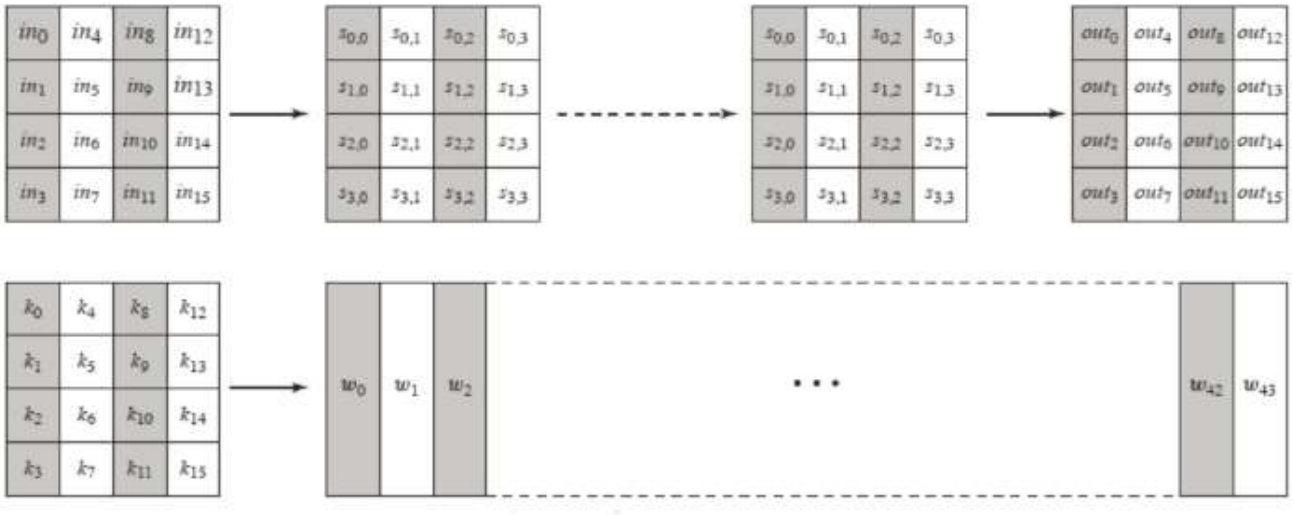


Figura 1. Estrutura de dados do AES

Fonte: STALINGS (2014, p. 105)

A encriptação por AES consiste em N rodadas, e o número delas depende do comprimento da chave: 10 rodadas para uma chave de 128 bits (16 bytes), 12 para uma chave de 192 bits (24 bytes) e 14 para uma chave de 256 bits (32 bytes). As primeiras N —1 rodadas consistem em quatro funções de transformação distintas: SubBytes, Shif¬tRows, MixColumns e AddRoundKey e a rodada final (N) contém apenas três, e há uma transformação inicial única (AddRoundKey) antes da primeira rodada, o que pode ser considerado Rodada 0. Cada transformação utiliza uma ou mais matrizes de tamanho 4 x 4 como entrada e produz outra de mesmo tamanho como saída [Stallings 2014].

2.2. CRIPTOGRAFIA RSA

O método de criptografia de chave pública (assimétrica) mais conhecido atualmente é o RSA. Inventado em 1978 por Ron Rivest, Adi Shamir e Len Adleman, que trabalhavam no Massachussets Institute of Technology (M.I.T.). As letras RSA correspondem as ini¬ciais dos seus criadores [Coutinho 1999]. RSA foi baseada na teoria dos números, em específixo na dificuldade computacional de fatorar um número inteiro em primos.

O conceito utilizado na criação das chaves é a função totiente de Euler $(\phi(n))$, a qual é definida como o número de inteiros positivos que são menores que n e relativamente primos de n, [Stallings 2014]. Como um número primo só pode ser dividido somente por si próprio e 1, fala-se então que para um primo p a função totiente deste número é: $\phi(p) = p - 1$. Então para dois números primos diferentes entre si p e q podemos mostrar que $n = pq$:

$$\phi(n) = \phi(pq) = \phi(p) \times \phi(q) = (p-1) \times (q-1)$$

Outro ponto importante a ser explicado é o teorema de Euler. Este que também é denominado de Pequeno Teorema de Fermat tem-se que se um número n é primo, então para cada e tal que $0 < e < n : a^{n-1} = 1$ mod n [Terada 2008].

Então através das teorias anteriores podemos fazer o cálculo de um par de chaves na criptografia RSA [Terada 2008]:

1. Calcular dois números primos denominados p e q e fazer o cálculo do seu produto $n = pq.$

2. Calcular um outro número s que seja relativamente primo a 0(n) e um inteiro r que satisfaça $\phi(n)$ mod $r.s = 1 \bmod (p-1)(q-1).$

3.Depois pode-se apagar os números p e q.

4. A chave secreta então é denominada $S = (s, n)$ e a chave pública $P = (r, n)$.

Para o envio de um texto legível a encriptação é feita em blocos de tamanho menor ou igual a $\log_2(n) + 1$ [Stallings 2014]. De tal forma que para um bloco de texto T e um texto cifrado C temos que:

$$C = T^e \bmod n$$
$$T = C^d \bmod n = (T^e)^d \bmod n = T^{ed} \bmod n$$

[Stallings 2014] cita ainda alguns requisitos que são necessários para que o algoritmo seja satisfatório seguindo a fórmula acima:

1. Consegue-se encontrar valores de e, d e n, de tal forma que T^{ed} mod $^{n=T}$ para todo $T < n$.
2. É possível calcular T^e mod n e C^d mod n para todos os valores de $T < n$.
3. Mesmo conhecendo e e n não é possível determinar d.

Para encontramos que T^{ed} mod $n = T$, a relação entre eles só é mantida se e e d forem inversos multiplicativos de $\bmod \phi(n)$. Logo, temos que a relação entre estes dois números é expressa como:

$$ed \bmod \phi(n) = 1$$
$$ed \equiv 1 \bmod \phi(n)$$
$$d \equiv e^{-1} \bmod \phi(n)$$

Tendo em vista toda a teoria, para um esquema RSA temos os seguintes compo-

nentes:

- dois números primos diferentes *p* e *q.*
- $n = pq$.
- *e* sendo $mdc(\phi(n), e) = 1; 1 < e < \phi(n)$
- $d \equiv e^{-1} (\mathrm{mod} \phi(n))$

3. METODOLOGIA

O aporte metodológico adotado nesse trabalho é de uma pesquisa comparativa, pois visa comparar os dois métodos criptográficos abordados em um banco de dados relacional, buscando definir qual método o método mais adequado para implementação de acordo com o tempo de criptografia. Então, para um melhor conhecimento das vantagens e des¬vantagens entre a criptografia RSA e AES foram feitos tanto testes computacionais.

3.1. CONFIGURAÇÃO DO BANCO

Para a comparação entre os dois modelos de criptografia apresentados, foram realizados testes computacionais em um computador com as determinadas configurações:

- Processador: Intel(R) Core(TM) i7-7500U CPU @ 2.70GHz
- Memória instalada (RAM): 8,00 GB
- Tipo de sistema: Sistema Operacional de 64 bits, processador com base em x64.

Utilizando a plataforma Microsoft® SQL Server® 2014 Express na versão de 64 bits, as análises de tempo foram feitas no SQL Server Management Studio. A versão escolhida para o trabalho a 12.0.2000.8 publicada na data de 25 de junho de 2014. O pacote escolhido para a implementação foi o Express With Tools, pois atende aos re¬quisitos necessários para as devidas finalidades propostas. Antes de instalar a aplicação faz-se necessário habilitar ou instalar o Microsoft.Net Framework 3.5 SP11 ou o Micro¬soft.Net Framework 4.02. O banco de dados analisado é disponibilizado gratuitamente no site da fabricante denominado AdventureWorks e foi configurado conforme os seguintes parâmetros:

1. Os arquivos baixados são de extensão .bak e colocado na pasta com o caminho C:\Program Files \ Microsoft SQL Server\ MS S QL12.SQLEXPRESS \ MS SQL \Backup;

2. Após fazer login no SQL Server Management Studio na barra de navegação es¬querda, clicando com botão direito em Bancos de Dados aparece a opção de Res¬taurar Banco de Dados.

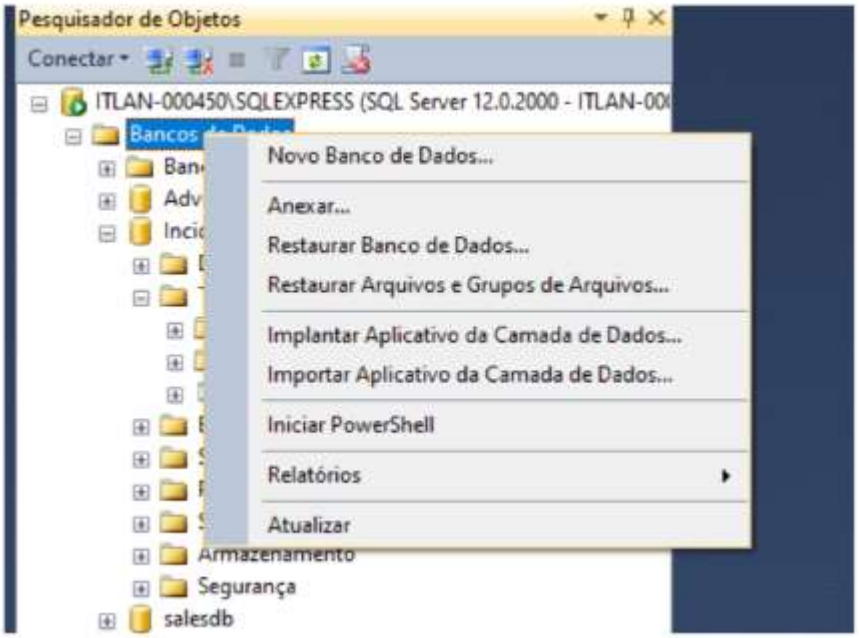


Figura 2. Restauração do BD

3. Clicando na opção de Restaurar Banco de Dados abre-se uma janela que tem a opção de escolher a origem de restauração. Neste caso escolhemos Dispositivo e clicando nos três pontos do lado direito, abre-se outra janela a qual possibilita escolher o arquivo baixado ao clicar na opção Adicionar.

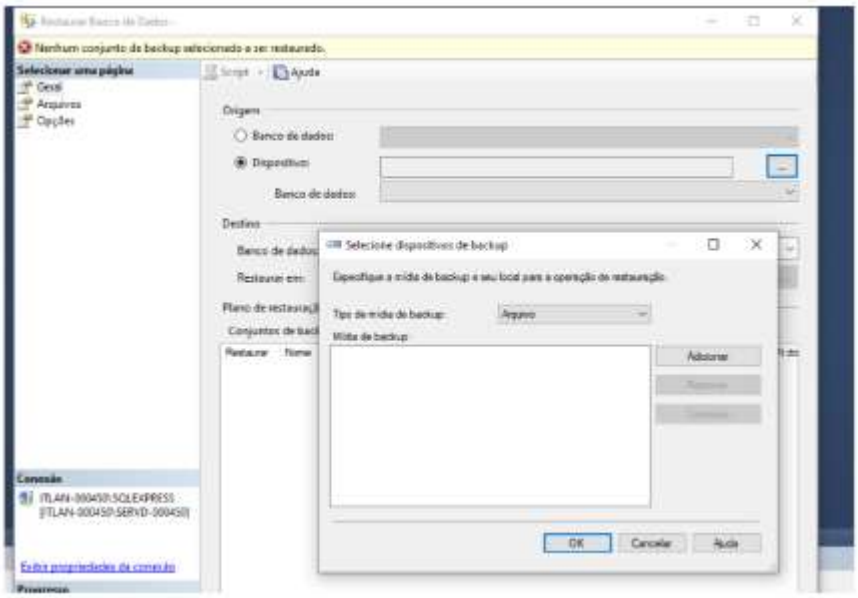


Figura 3. Escolha da origem

4. Depois da escolha e confirmação do arquivo desejado, o banco de dados irá estar presente no servidor como mostra a Figura 4 abaixo:

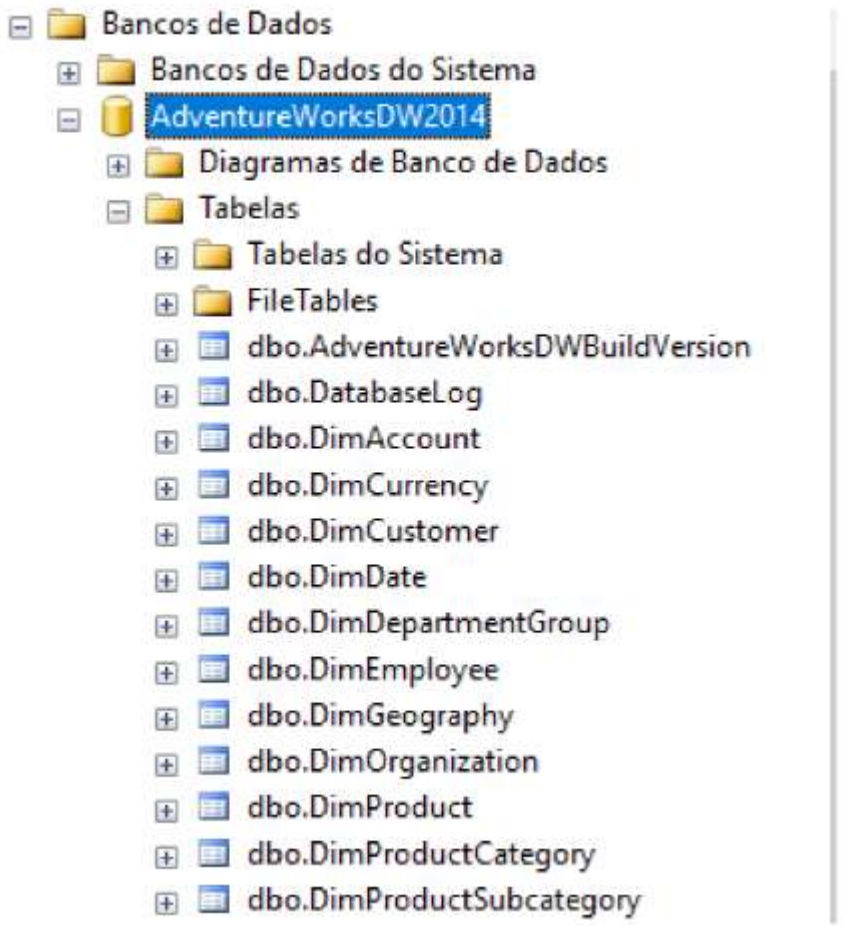


Figura 4. Banco de Dados implementado

Com estes passos feitos, o banco de dados está pronto para a implementação das criptografias.

3.2. IMPLEMENTAÇÃO

Na documentação do SQL Server existem funções criptográficas que são responsáveis pela criação do par de chaves e também na implementação da criptografia. Para iniciar o processo de criptografia das colunas da tabela faz-se necessário seguir uma hierarquia. Primeiro cria-se uma Master Key no nível de instância caso não tenha. Utilizando então o método CREATE MASTER KEY, através de uma Nova Consulta (Query), definindo uma senha a ser utilizada como demonstrado abaixo:

```
USE master;
GO
IF NOT EXISTS(
     SELECT *
     FROM sys.symmetric_keys
     WHERE name = '##MS_ServiceMasterKey##')
BEGIN
     CREATE MASTER KEY ENCRYPTION BY
     PASSWORD = 'MSSQLSerivceMasterKey'
END;
GO
```

Após isto foram utilizadas as funções CREATE ASYMMETRIC KEY e CREATE SYMMETRIC KEY, as quais são responsáveis pelas criações das chaves assimétrica RSA e simétrica AES. Em relação ao tamanho utilizado nas chaves, para a criptografia RSA o tamanho escolhido foi de 512 bits e para o AES 256 bits.

Com a criação das chaves, com as funções de encriptação: ENCRYPTBYKEY e ENCRYPTBYASYMKEY, foram feitas as criptografias de uma coluna específica deno¬minada Phone disponível na tabela DimCostumer do banco apresentado. Então a cripto¬grafia da coluna por chave assimétrica possui o seguinte código:

```
USE [AdventureWorksDW2014]
GO
     CREATE ASYMMETRIC KEY AsymKey
     WITH ALGORITHM = RSA_512
     ENCRYPTION BY PASSWORD = 'T33sT@nD!'
     ;
END
GO
ALTER TABLE DimCustomer
ADD PCrip varbinary(MAX) NULL
GO
UPDATE A
SET PCrip = ENCRYPTBYASYMKEY(ASYMKEY_ID('AsymKey'), Phone)
FROM dbo.DimCustomer AS A;
GO
```

Para a chave simétrica AES, o código apresentado é de certa forma parecido com o anterior com a ressalve que para o processo de encriptação é necessário a criação de um certificado anteriormente:

```
USE [AdventureWorksDW2014]
CREATE CERTIFICATE CertCrip
WITH SUBJECT = 'CertificadoCriptografia';
GO
CREATE SYMMETRIC KEY SimKey
WITH ALGORITHM = AES_256
ENCRYPTION BY CERTIFICATE CertCrip;
GO
ALTER TABLE DimCustomer
ADD SimCrip varbinary(MAX) NULL
GO
OPEN SYMMETRIC KEY SimKey
    DECRYPTION BY CERTIFICATE CertCrip;
        GO
UPDATE A
SET SimCrip = ENCRYPTBYKEY(Key_GUID('SimKey'), Phone)
FROM dbo.DimCustomer AS A;
GO
```

Para efeitos de visualização as colunas criptografas aparecem de tal forma quando selecionadas na aplicação:

Resultados | Mensagens

	FirstName	Phone	SimCrip
1	Jon	1 (11) 500 555-0162	0x00930B8017A96145BF1452A2719973610100000C233E9...
2	Eugene	1 (11) 500 555-0110	0x00930B8017A96145BF1452A27199736101000000080D9E7...
3	Ruben	1 (11) 500 555-0184	0x00930B8017A96145BF1452A2719973610100000001E05D8...
4	Christy	1 (11) 500 555-0162	0x00930B8017A96145BF1452A27199736101000000A674686...
5	Elizabeth	1 (11) 500 555-0131	0x00930B8017A96145BF1452A27199736101000000A700440...

Figura 5. Criptografia assimétrica no banco

Resultados | Mensagens

	FirstName	Phone	SimCrip
1	Jon	1 (11) 500 555-0162	0x00930B8017A96145BF1452A2719973610100000C233E9...
2	Eugene	1 (11) 500 555-0110	0x00930B8017A96145BF1452A27199736101000000080D9E7...
3	Ruben	1 (11) 500 555-0184	0x00930B8017A96145BF1452A2719973610100000001E05D8...
4	Christy	1 (11) 500 555-0162	0x00930B8017A96145BF1452A27199736101000000A674686...
5	Elizabeth	1 (11) 500 555-0131	0x00930B8017A96145BF1452A27199736101000000A700440...

Figura 6. Criptografia simétrica no banco

4. COMPARATIVO E RESULTADOS

A chave assimétrica, RSA com tamanho de 512 bits, foi denominada como AsymKey e a simétrica, AES com 256 bits, como SimKey. Para a utilização das funções crip-tográficas foram criadas 10 novas colunas que serão contadas como rodadas de teste. Cada encriptação em uma nova coluna a partir de uma coluna existe tanto pela função simétrica quanto assimétrica será contabilizada como uma rodada.

Para analisar o tempo de criptografia de cada rodada foram habilitadas as estatísticas de Cliente. Estando disponível através do comando Shift + Alt + S ou na aba Consulta. Com esta função habilitada é possível visualizar as estatísticas de tempo de execução de cada consulta feita, ou seja, cada nova coluna criptografada criada.

Rodada de teste	**Chave**	
	RSA	AES
1	1.798	645
2	1.138	670
3	1.152	606
4	1.194	613
5	1.178	615
6	1.464	594
7	2.154	609
8	1.151	644
9	1.178	690
10	1.222	730
Média (apr.)	1.363	641

Tabela 1. Tempo para criptografar (em ms)

Com uma diferença média de 722 ms a criptografia AES mostrou-se mais rápida na implementação. O que a torna melhor caso tenha-se a necessidade de menos tempo em implementação não levando em conta a segurança por trás de cada um dos algoritmos.

Figura 7. Gráfico comparativo do tempo

Figura 7. Gráfico comparativo do tempo entre os dois métodos. Enquanto a criptografia RSA possuir um tempo maior de proces-samento ainda apresenta certas oscilações entre as rodadas de testes.

5. CONSIDERAÇÕES FINAIS

A criptografia puramente assimétrica é muito mais lenta do que as cifras simétricas (como o AES), devido a isso aplicações atuais usam criptografia híbrida: as operações de chave pública são executadas apenas para criptografar (e trocar) uma chave de criptografia pelo algoritmo simétrico que está sendo executado para ser usado para criptografar a mensa¬gem real.

Porém, caso tenha que se fazer a escolha então de qual algoritmo deve ser imple¬mentado tem que considerar dois fatores: o poder computacional necessário para crip¬tografar/descriptografar os dados; e se tem-se a necessidade de fazer uso de uma chave síncrona ou assíncrona. Ao considerar o poder computacional, especialmente quando há a necessidade de codificar grandes volumes de dados em equipamentos com capacidade limitada, optar pelo AES tem sido a escolha mais popular pois como visto anteriormente possui um tempo mais rápido de processamento.

REFERÊNCIAS

Coutinho, S. C. (1999). The mathematics of ciphers: number theory and RSA crypto-graphy. AK Peters/CRC Press.

Kim, D. and Solomon, M. G. (2014). Fundamentos de segurança de sistemas de informação. Rio de Janeiro: LTC.

Stallings, W. (2014). Criptografia e segurança de redes . Pearson Educacion.

Terada, R. (2008). Segurança de dados: criptografia em rede de computador. Editora Blucher.

Capítulo 5

10.37423/220606220

MODELO DE DETECÇÃO DE DISPOSITIVOS IOT ATRAVÉS DE DENAT

Sergio Lucio Nunes da Silva — *Universidade Federal de Mato Grosso*

Jadson Matheus Bezerra de Lima — *Universidade Federal de Mato Grosso*

Gabriel Jaber Aguiar — *Universidade Federal de Mato Grosso*

Lucas santos de almeida — *Universidade Federal de Mato Grosso*

Gracyeli Santos Souza Guarienti — *Universidade Federal de Mato Grosso*

Joyce Aline de Oliveira Marins — *Universidade Federal de Mato Grosso*

Resumo: O aumento e a popularização dos dispositivos IoT associados com vulnerabilidades criadas por falha de fabricação ou negligência do usuário têm tornado esses dispositivos uma interface muito atraente para ataques. O objetivo dos atacantes geralmente é cons- truir botnets a serem utilizadas em ataques de negação de serviço - DDoS. Neste trabalho é reproduzido e avaliado um modelo de detecção de dispositivos potencialmente vulne- ráveis que podem estar presentes em ambiente domiciliar dos clientes de provedores de Internet. O modelo proposto pode ser instalado como um intermediário entre a rede do- miciliar dos clientes e o ponto de conexão da provedora. Os resultados apresentam a comparação da eficiência do algoritmo de treino do modelo, que é o LGBM, com um algoritmo proposto recentemente, Neural Decision Forest, que une o algoritmo de florestas aleatórias às redes neurais profundas.

Palavras-chave: IoT, LGBM, Neural Decision Forest, Modelo de detecção.

CAPÍTULO 1

INTRODUÇÃO

Os dispositivos IoT podem coletar uma diversidade enorme de dados que variam desde temperatura e umidade de ambientes até informações muito pessoais como bati- mentos cardíacos e hábitos cotidianos de seus usuários (Meneghello et al. 2019). Por conta da característica de alta conectividade, é comum que eles sejam usados tam- bém como porta de entrada para ataques maiores nas redes em que estão conectados (Meneghello et al. 2019).

Os dispositivos IoT têm sido um constante alvo para criação de redes de dispositi- vos botnets que serão utilizadas para efetuar ataques DDoS (Hummel et al. 2020, Wang et al. 2018). Um ataque DDoS acontece quando o atacante busca tornar determinado ser- viço web indisponível aplicando uma carga excessiva de carregamento ou até deixando o serviço totalmente fora do ar, e usa diferentes fontes para isso. Em 2020, o número de ataques DDoS registrados passou de 10 milhões pela primeira vez na história. Entre as empresas que prestam serviços web, 71% consideraram os ataques DDoS como a maior ameaça do ano (Hummel et al. 2020).

Os dispositivos IoT são alvos de ataques que comprometem toda a cadeia de agentes. Este tipo de ataque pode comprometer desde a pessoa que adquiriu o dispositivo e teve sua rede invadida, até o serviço que sofreu o DDoS e ficou fora do ar, passando por um agente importante, a companhia de telecomunicações (TELCO), fornecedora do serviço de Internet que tem sua infraestrutura comprometida por tal fato.

Neste trabalho, é reproduzido um modelo de detecção de dispositivos IoT em ambiente domiciliar baseado na análise de fluxos da rede usando deNAT. O deNAT é o processo reverso do NAT, que traduz endereços IP de uma rede interna através de um roteador ou firewall para um endereço IP válido na Internet. A proposta é que a telco execute o modelo na sua infraestrutura usando os dados gerados por esse modelo para detectar os dispositivos vulneráveis, e assim, de maneira ativa, atuar na prevenção de ataques aos dispositivos reconhecidamente vulneráveis (Meidan et al. 2020).

Para atingir o objetivo de detecção o modelo utiliza dados da rede organizados através da funcionalidade NetFlow, presente em diversos roteadores Cisco, que tem a capacidade de coletar o tráfego da rede IP conforme ela entra e sai de uma interface. Esses dados são inseridos em um algoritmo LGBM para treinamento de um classificador que, na fase de predição, de acordo com o fluxo,

será capaz de detectar qual é o dispositivo IoT que está rodando no cliente, sabendo o dispositivo, é possível verificar se trata-se de um dispositivo vulnerável ou não.

Esse trabalho busca contribuir com a solução do problema propondo:

- a reprodução do modelo de detecção via LGBM proposto por (Meidan et al. 2020);
- o desenvolvimento de um modelo basedo em NDF - Neural Decision Forest;
- a comparação da acurácia do modelo treinado com o LGBM e o modelo treinado com NDF; e
- uma metodologia para implantação do modelo;

CAPÍTULO 2

MATERIAIS E MÉTODOS

O método utilizado envolve diversas etapas, que vão desde a construção de um setup de coleta dos dados até a validação do modelo de classificação. A arquitetura do setup de coleta de dados para treinamento do modelo é simples e pode ser dividida em algumas etapas: coleta dos dados, organização dos dados, treinamento e testes.

2.1 COLETA DOS DADOS

Na etapa de coleta os dispositivos se conectam a um access point, e este, por sua vez, se conecta ao roteador que liga a rede local à Internet. Para coletar os dados, o roteador deve conter um dos protocolos de organização de tráfego da rede por fluxo, como o NetFlow, da Cisco. Esses dados são coletados por um detector local na rota de saída do roteador para a conexão com a rede da telco.

2.2 ORGANIZAÇÃO E PRÉ-PROCESSAMENTO

Após a coleta dos dados organizados em fluxo pelo protocolo Netflow, os dados são colocados de maneira estruturada em formato CSV através de um script. Essa etapa é fundamental para viabilizar e tornar os dados compreensíveis para os algoritmos de aprendizado de máquina. Já dentro da etapa de treinamento do modelo e de testes do mesmo, deve ser feito o pré-processamento dos dados estruturados do arquivo CSV, etapa indispensável para o bom desempenho dos algoritmos de classificação.

2.3 TREINAMENTO DO MODELO

O arquivo contendo os fluxos disponibilizado no artigo (Meidan et al. 2020) contém de maneira explicita em uma das colunas a separação entre os dados de treino e os dados de teste. São mais de um milhão e meio de entradas no total, das quais, 82% são dados para treino e 18% são destinados para os testes. Ao fim do treinamento, o modelo é salvo, carregado e testado com as entradas destinadas para teste, como explicado anteriormente. Para criar os modelos de detecção são utilizados dois algoritmos.

O LGBM, Light Gradient Boost Machine, é um algoritmo GBDT (Gradient Boost Deci- sion Tree) criado com o objetivo de ter melhor desempenho ao lidar com dados volumo- sos e muitas features (ie.,

características dos dados, como colunas de uma tabela), se comparado a algoritmos como XGBoost, sendo mais rápido e menos exigente computa- cionalmente.

O algoritmo NDF, Neural Decision Forest, busca otimizar os resultados de um modelo unindo dois métodos muito utilizados em aprendizado de máquina, as árvores de decisão e as redes neurais profundas (Kontschieder et al. 2015).

Visando a qualidade, eficiência do modelo e capacidade de detecção da maior varia- bilidade possível de dispositivos, é necessária uma amostra heterogênea de dispositivos na fase inicial de coleta de dados a fim de que os dados dos mesmos sejam considerados na fase de treinamento dos modelos.

2.4 EXPLORANDO O DATASET

O dataset disponibilizado por (Meidan et al. 2020) possui 24 atributos (colunas) e 1.564.789 registros, os dados foram coletados em um período de trinta e seis dias, em um setup montado em laboratório.

Conforme a Tabela 2.1, é visto que dos doze dispositivos monitorados para a coleta de dados, cinco não são dispositivos IoT e sete são, isso é importante e foi feito propo- sitalmente, a fim de gerar dados para treino e validação que fossem compatíveis com os cenários domésticos, uma vez que esse é o cenário mais comum, dessa forma, o modelo já seria treinado com dados de tráfego de dispositivos não IoT.

Apesar das vinte e quatro features presentes no dataset, as features que de fato serão utilizadas para treinar o modelo são doze, descritas a seguir:

1. FIRST_SWITCHED: Tempo de atividade do sistema em que o primeiro pacote do fluxo foi comutado.
2. IN_BYTES: Contador de bytes de entrada.
3. IN_PKTS: Contador de número de pacotes de entrada.
4. IPV4_DST_ADDR: Endereço IPV4 do destino.
5. L4_DST_PORT: Número da Porta TCP/UDP de destino.
6. L4_SRC_PORT: Número da Porta TCP/UDP de origem.
7. LAST_SWITCHED: Tempo de atividade do sistema em que último pacote foi comu- tado.
8. SRC_TOS: Tipo de byte de serviço quando há um interface de entrada.
9. TCP_FLAGS: Contador de todos os sinalizadores TCP vistos para este fluxo.

Dentre essas features, diversas delas não eram compreensíveis para os algoritmos a serem testados, por exemplo, a feature IPV4_DST_ADDR tem o formato IP (ie. 127.0.0.1) que notavelmente é não numérico e precisa ser transformado. Dessa forma, foi necessário um pré-processamento diferente

para cada algoritmo. O pré-processamento se resumiu a transformar as features categóricas em numéricas, uma vez que o dataset não possui registros com dados dessas features que sejam faltosos, nulos ou algo semelhante.

Tabela 2.1: Tabela com categoria e tipo dos dispositivos monitorados na coleta de dados.

Dispositivos monitorados			
Classe	Dispositivo	Categoria	Tipo
0	webcam.Samsung.SNH-1011N	IoT	IoT
1	laptop.Dell.Latitude_7400	Non-IoT	laptop
2	light_bulb.TP_Link.LB130	IoT	IoT
3	laptop.Dell.Latitude_E6430	Non-IoT	laptop
4	speaker.Sonos.One	IoT	IoT
5	doorbell.Amazon.Ring	IoT	IoT
6	access_point.TP_Link.TL-WA901ND	Non-IoT	access_point
7	webcam.Amcrest.IPM-721W	IoT	IoT
8	smartphone.Samsung.Galaxy_Note_5	Non-IoT	smartphone
9	socket.Wemo.Insight	IoT	IoT
10	smartphone.Samsung.Galaxy_Note_4	Non-IoT	smartphone
11	streamer.Amazon.Fire_TV_Stick	IoT	IoT

2.5 MÉTRICAS DE AVALIAÇÃO

As métricas de avaliação estão presentes principalmente no capítulo de resultados e são usadas como um parâmetro a partir do qual é feita a comparação de modelos diferentes, levantando hipóteses sob o desempenho de cada um. As métricas usadas neste trabalho foram escolhidas com base no artigo referência (Meidan et al. 2020) e sua definição forma também está presente em (Lipton et al. 2014).

A métrica acurácia consiste na porcentagem de acertos do modelo, e é calculada da seguinte maneira:

$$\text{Acurácia} = \frac{\text{Verdadeiro Positivo} + \text{Verdadeiro Negativo}}{\text{Total}}, \qquad (2.1)$$

em que Verdadeiro Positivo e Verdadeiro Negativo indicam a quantidade de acertos do modelo binário para as classes positiva e negativa. A Acurácia é número real entre 0 (zero) e 1 (um). Quanto maior o valor (mais próximo de 1), melhor é o desempenho do classificador.

A métrica denominada Precisão é calculada levando em conta uma relação entre ver- dadeiros positivos e falsos positivos, da seguinte maneira:

$$\text{Precisão} = \frac{\text{Verdadeiro Positivo}}{\text{Verdadeiro Positivo} + \text{Falso Positivo}}, \quad (2.2)$$

em que Falso Positivo indica a quantidade de elementos da classe negativa que foram classificados com positivos. Dentro do nosso contexto significa que quanto mais próximo de zero, maior foi a quantidade de vezes que o modelo classificou um dispositivo como pertencente a classe referência e na verdade ele pertencia a outra classe.

A revocação é definida como uma relação entre os verdadeiros positivos e falsos negativos, de tal forma que:

$$\text{Revocação} = \frac{\text{Verdadeiro Positivo}}{\text{Verdadeiro Positivo} + \text{Falso Negativo}}, \quad (2.3)$$

em que Falso Negativo corresponde ao número de elementos da classe negativa que foram classificadas erroneamente. Quanto mais próximo de zero, maior a quantidade de vezes que modelo classificou um dispositivo como pertencente a outra classe e na verdade ele pertencia a classe de referência.

Por fim, F1-score é definida como a média entre Precisão e Revocação, da seguinte maneira:

$$\text{F1-score} = \frac{2}{\frac{1}{\text{Revocação}} + \frac{1}{\text{Precisão}}}. \quad (2.4)$$

CAPÍTULO 3

EXPERIMENTOS E RESULTADOS

Este Capítulo apresenta os resultados experimentais da aplicação dos algoritmos LGBM e NDF. Comparando os resultados atingidos por cada um e avaliando as métri- cas utilizadas e o impacto delas. (Kontschieder et al. 2015). A análise dos modelos levou em conta a identificação dos atributos mais importantes e a localização de viés no modelo original. O impacto desse viés foi avaliado nesse trabalho e é um avanço em relação ao trabalho original(Meidan et al. 2020).

3.1 CONSIDERAÇÕES INICIAIS SOBRE O ENVIESAMENTO DO MO- DELO DA LITERATURA

Na análise dos atributos utilizados por (Meidan et al. 2020), nota-se que o uso do atributo IPV4_DST_ADDR pode ser questionado, uma vez que essa feature representa o endereço IP do destino. É natural que um dispositivo IoT da marca X tenha como IP de destino, na maioria das vezes, o servidor da marca X. Muitos dos dispositivos IoT acabam tendo como servidor um IP que tem prefixo da Amazon Web Services, por exemplo, ou outra plataforma semelhante, o que é natural, uma vez que a empresa fornecer diversas soluções para esse tipo de serviço. Contudo, mesmo com prefixo igual, esses endere- ços são diferentes, mantendo o risco de enviesamento do modelo. Por isso, o segundo experimento desta seção buscou avaliar os resultados dos modelos sem esse atributo, considerando apenas os outros dados de tráfego da rede, isso valida se realmente os al- goritmos conseguem encontrar uma maneira de classificar os dispositivos com dados de fora da rede doméstica.

3.2 EXPERIMENTO 1

3.2.1 MODELO LGBM

O modelo treinado com LGBM contendo o atributo de IP do destino será chamado de modelo L1. Para o treinamento do modelo L1, é feito o carregamento do dataset, a separação dos dados em dados de treino e teste conforme o rótulo, aplicado o método de ajuste que calcula os parâmetros iniciais sobre os dados de treino, e por fim, aplicado

Classification Report ----------

	precision	recall	f1-score	support
0.0	0.00	0.00	0.00	2505
1.0	0.55	0.94	0.70	3156
2.0	0.07	0.01	0.02	946
3.0	0.00	0.00	0.00	0
4.0	0.95	1.00	0.98	19520
5.0	0.81	0.79	0.80	12490
6.0	0.26	0.16	0.20	2429
7.0	0.98	0.93	0.96	62867
8.0	0.78	0.99	0.87	14832
9.0	0.99	0.90	0.94	30619
10.0	0.95	1.00	0.97	90709
11.0	1.00	0.97	0.98	46054
accuracy			0.94	286127
macro avg	0.61	0.64	0.62	286127
weighted avg	0.93	0.94	0.93	286127

Figura 3.1: Relatório de classificação - L1.

o método de transformação. Nos dados de teste, ou previsão, não deve ser utilizado o método de ajuste, pois durante o treinamento o modelo deve encontrar os devidos parâ- metros. Entretanto, é preciso aplicar o método de transformação para que os atributos categóricos possam ser lidos pelo modelo.

Após isso, é necessário reduzir as colunas do dataset, deixando apenas as que de fato interessam para o treinamento do modelo. Esse é o pré-processamento necessário. É preciso separar os dados de entrada e de saída, sabendo que a saída deve ser um modelo de dispositivo detectado. Assim o modelo já pode ser treinado.

Os parâmetros do LGBM foram configurados da seguinte forma: o número de folhas da árvore foi fixado em 32 e as iterações máximas em 30, o restante dos parâmetros ficaram definidos de acordo com o padrão da implementação da função. O modelo L1 leva poucos segundos para ser treinado e testado, e ao fim dos testes é obtida uma acurácia de 93,91%. A Figura 3.1 apresenta o relatório de classificação do modelo L1. Para sabermos qual classe representa cada dispositivo, basta consultarmos a tabela 2.1. É possível descartar os resultados do dispositivo da classe 3, pois não há entradas nos dados de teste para ele, como visto na coluna support.

Os resultados do dispositivo 0 são interessantes, o modelo L1 não classificou-o corre- tamente nenhuma vez, por isso não há dados de precisão e revocação para ele. A matriz de confusão mostra os resultados de verdadeiro positivo, verdadeiro negativo, falso posi- tivo e falso negativos para cada atributo. Observando a Figura 3.2, é visto que 93% das vezes ele foi classificado como sendo o dispositivo 1, ou seja, falso negativo, e em sua classe não houve nenhum caso de falso positivo. O resultado da classificação do disposi- tivo 0 afetou os resultados do dispositivo 1, já que esse dispositivo teve um índice alto de falsos positivos vindos de 0, classificados como 1, isso reduziu seu f1-score. O dispositivo

```
Confusion Matrix -----------------
[[    0  2335   100    55    12     0     0     0     0     0     0     3]
 [    0  2952    39    20    68    39     6    14    14     1     3     0]
 [    0     1    12   136   371   305    83    26    12     0     0     0]
 [    0     0     0     0     0     0     0     0     0     0     0     0]
 [    0     0     0     0 19518     0     0     0     0     0     0     2]
 [    0    32     7    16   366  9899   819   556   760    34     0     1]
 [    0     0    18    12    96  1763   393    85    57     5     0     0]
 [    0     0     0     0     0     0     0 58665   457     0  3745     0]
 [    0     0     0     1     9    21     2    10 14750     0     0    39]
 [    0     0     0     6     8   166   184   366  2371 27516     0     2]
 [    0     0     0     0     0     0     2     4    47   358 90298     0]
 [    0     0     0     0     0     0     0    27   467    14   848 44698]]
```

Figura 3.2: Matriz de confusão - L1.

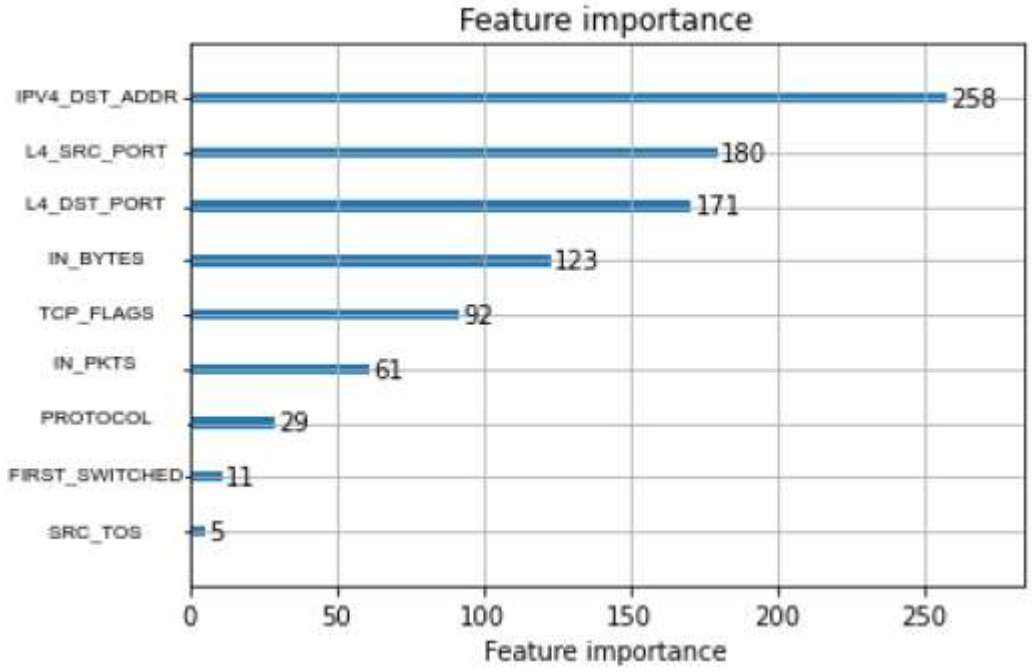


Figura 3.3: Importância das features - L1.

2 possui poucas entradas, o que pode ter contribuído para o fracasso do modelo L1 em classificá-lo. Por ser um dispositivo de rede, era esperado que a classificação do dispo- sitivo 6 não fosse tão bem sucedida, os dados de tráfegos variam muito, o que dificulta a definição de padrões.

Dessa forma, considerando que os menores índices de acerto correspondem ao dis- positivos que possuem as menores quantidades de entradas, podemos supor que há uma relação direta entre a quantidade de dados trafegados por um dispositivo e as chances de conseguir detectá-lo corretamente, isto é, quanto mais dados um dispositivo trafega, mais fácil detectá-lo, classificando-o corretamente.

Para o algoritmo LGBM, conforme a Figura 3.3, é vista a importância das features. Cabe destacar que a feature de IP do destino, é justamente a mais importante, com al- gumas dezenas de pontos à frente da segunda e terceira colocada, que são as portas TCP/UDP de origem e destino, respectivamente.

3.2.2 MODELOS NDF

As diferenças da fase pré-treinamento do modelo NDF são as seguintes: não é utili- zado método de ajuste e transformação para transformar variáveis categóricas em numé- ricas, no lugar disso, é definido quais são as variáveis numéricas e categóricas e qual é a variável que deve ser predita. A partir dessa configuração, a responsável pela transforma- ção dessas variáveis e pela criação dos dicionários é a função StringLookup da biblioteca Keras, que traduz um conjunto de strings arbitrárias em uma saída de número inteiro por meio de uma pesquisa de vocabulário baseada em tabela.

Com as variáveis devidamente transformadas para serem utilizadas no treinamento do modelo, segue- se o próximo passo. O algoritmo NDF possui dois tipos de modelos, o modelo de árvore e o modelo de floresta, para ambos o treinamento foi realizado com 30 épocas e taxa de aprendizado de 0.00001. O modelo de árvore, que chamaremos a partir de agora de modelo A, resulta em um único modelo de árvore de decisão neural que utiliza todos os recursos de entrada, nesse modelo, os parâmetros são configurados da seguinte forma: número de arvores = 10; profundidade = 10. No modelo de floresta, que será chamado a partir de agora de modelo F, uma floresta de árvores de decisão é treinada e cada árvore usa 50% dos recursos de entrada, e isso pode ser alterado, o número de árvores foi definido como 25 e a profundidade foi reduzida para 5.

É válido ressaltar que o tempo de treinamento de um modelo a partir do NDF é muito mais lento do que a partir do LGBM, uma vez que o NDF é baseado em redes neurais profundas, naturalmente, também consome mais recursos e dependendo do caso, o uso de GPU é recomendável. Rodando a

partir do Google Colab, cada época de treinamento do NDF demora entre 10 e 15 minutos. Com as configurações de profundidade e número de árvores conforme supracitado, o treinamento de cada modelo pode levar de 5 a 7 horas.

MODELO A1

Após 30 épocas de treinamento, o modelo A1 foi avaliado e sua acurácia chegou a 81,18%, o que não diz muito, pois o modelo pode ser tendencioso a determinadas maneiras de classificar. Os gráficos da Figura 3.6 mostram as curvas de aprendizado de acurácia e de loss ao longo das épocas de treinamento. É observado através da Figura

3.4 que inicialmente a acurácia é baixa, mas que sobe rapidamente, e antes mesmo da metade do treinamento, já ultrapassa 90%. A curva da loss no treinamento é diferente, e pode ser vista na Figura 3.5. Ao longo do treinamento a loss cai a cerca de 65% de seu valor inicial.

Apesar dos resultados positivos no treinamento, a avaliação das métricas dos testes através da Figura 3.7 e da matriz de confusão da Figura 3.8, mostra de forma mais clara o desempenho do modelo A1 na classificação dos dispositivos. Sabendo que a diagonal principal da matriz de confusão mostra os resultados classificados corretamente, e vendo o f1-score, notamos que para 3 dispositivos (4, 7, 10) o modelo A1 tem acurácia próxima a 100%. Além disso, o dispositivo 0 teve um ótimo resultado de classificação, com precisão 100% e revocação de 87%, ambos os resultados bem diferentes do modelo L1, treinado por LGBM.

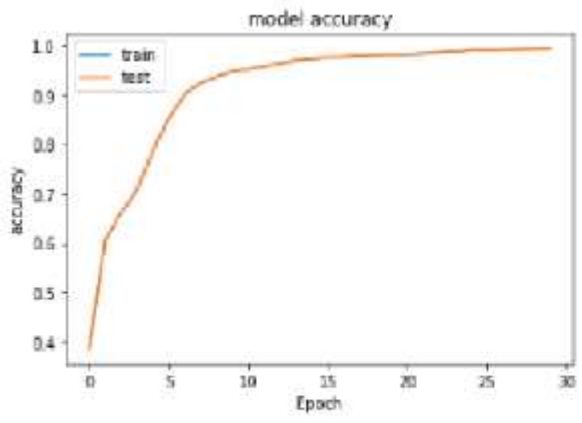


Figura 3.4: Acurácia

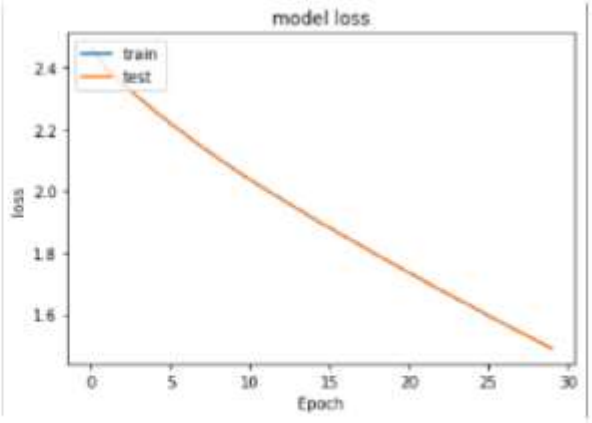


Figura 3.5: Loss

Figura 3.6: Gráfico de aprendizado - Modelo A1.

Contudo, para o restante dos dispositivos o resultado não é nada satisfatório, inclusive, os dispositivos 2, 6, 8 não tem nenhum resultado positivo de classificação, como pode ser visto na matriz de confusão da Figura 3.8. Como 3 dispositivos foram classificados de maneira incorreta para a maioria das entradas, isso afetou o resultado da classificação de outros dispositivos que tiveram bons resultados na precisão, mas péssimos resultados em revocação, ou vice-versa, como é o caso dos dispositivos 5, 9 e 1. Essas falhas só são perceptíveis ao avaliar os resultados com mais cuidado. Conforme visto na Figura 3.7, a média não ponderada por rótulo (macro avg), mostra um resultado de 50%, bem diferente da acurácia geral de 81%.

MODELO F1

Este modelo é composto por várias árvores ao invés de uma única, e diferente do modelo A1, aqui cada árvore utilizará 50% dos recursos de entrada, sendo assim, os resultados na fase de treinamento serão mais variados. O modelo foi treinado com 25 árvores com profundidade 5 e as mesmas 30 épocas do modelo anterior.

A Figura 3.11 mostra os resultados do treinamento. A curva de acurácia na fase de treinamento vista na Figura 3.9 é levemente diferente da Figura 3.4, que mostra a curva da acurácia do treinamento do modelo A1. No treinamento desse modelo a acurácia demora um pouco mais pra subir, mas chega a bons índices.

Já na fase de testes, comparando o f1-score, o resultado obtido é semelhante para os mesmos dispositivos. Com exceção de dois casos, o dispositivo 0, que teve um bom resultado de classificação no modelo A1, agora teve todos os seus resultados de clas- sificação feitos de maneira incorreta; e o dispositivo 5 que teve resultados medianos no modelo A1, mas, no modelo F1, teve resultado satisfatório. Ademais, os modelos não diferem de maneira absurda, os erros e acertos se distribuem e no geral os resultados são muito semelhantes.

Classification Report ----------

	precision	recall	f1-score	support
0	1.00	0.87	0.93	2505
1	1.00	0.30	0.47	3156
2	0.00	0.00	0.00	946
3	0.00	0.00	0.00	0
4	0.99	1.00	1.00	19520
5	0.39	0.89	0.54	12490
6	0.00	0.00	0.00	2429
7	1.00	1.00	1.00	62867
8	0.38	0.00	0.00	14832
9	0.46	1.00	0.63	30619
10	1.00	1.00	1.00	90709
11	1.00	0.31	0.47	46054
accuracy			0.81	286127
macro avg	0.60	0.53	0.50	286127
weighted avg	0.87	0.81	0.79	286127

Figura 3.7: Relatório de classificação - Modelo A1.

Confusion Matrix ----------

```
[[ 2169     0     0     0     0   298     0     0     0     0     0    38]
 [    0   958     0     0     0     1     0     0     0  2157    40     0]
 [    0     0     0     2     0   123     0     0     0   821     0     0]
 [    0     0     0     0     0     0     0     0     0     0     0     0]
 [    0     0     0     0 19513     1     0     0     0     6     0     0]
 [    0     0     0     0     0 11135     0     2     7  1346     0     0]
 [    0     0     0     0     0   919     0     5     0  1505     0     0]
 [    0     0     0     0     0     0     0 62867     0     0     0     0]
 [    0     0     0     0     0    17     0     0     5 14810     0     0]
 [    0     0     0     0     0    13     0     0     1 30605     0     0]
 [    0     1     0     0     2     0     0     0     0     2 90704     0]
 [    0     0     0     0   103 16385     0     1     0 15237     0 14328]]
```

Figura 3.8: Matriz de Confusão - Modelo A1.

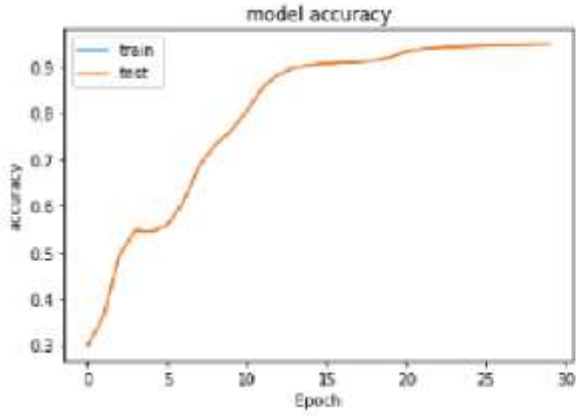


Figura 3.9: Acurácia

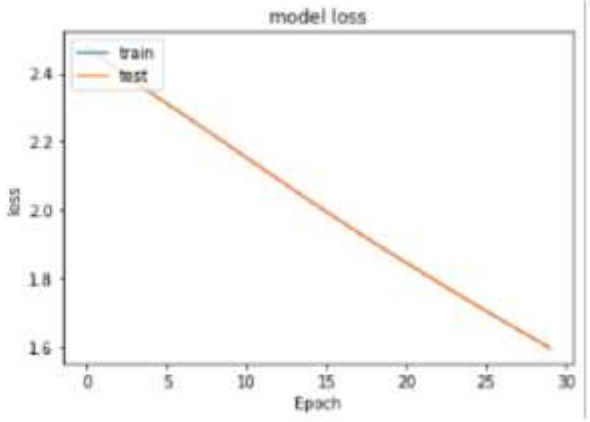


Figura 3.10: Loss

Figura 3.11: Gráfico de aprendizado - Modelo F1.

3.3 EXPERIMENTO 2

Neste experimento, foram avaliadas as diferenças nos resultados dos modelos treina- dos por ambos os algoritmos na ausência do atributo IPV4_DST_ADDR. A ideia é validar a capacidade dos algoritmos de encontrar um padrão nos dados de tráfego de rede que seja capaz de gerar um modelo com nível aceitável de acerto na classificação dos dispo- sitivos. Para tal tarefa, as alterações nos códigos de ambos os modelos foram mínimas, cabendo apenas a retirada da coluna que continha o atributo no dataset.

3.3.1 MODELO LGBM

Os resultados do modelo L2, treinado a partir do algoritmo LGBM para o experimento 2, são avaliados a partir da comparação com os resultados do experimento 1 usando o relatório de classificação, a importância das features e a matriz de confusão.

O relatório de classificação e a matriz de confusão do modelo L2 podem ser vistos nas Figuras 3.14 e 3.15, respectivamente. Através dos relatórios de classificação dos dois modelos, é notável que os dispositivos 0 e 2 possuem apenas erros na sua classifi- cação e os dispositivos 5, 6, 7, 8 e 9 têm o f1-score drasticamente afetado, tornando sua classificação pouco eficaz.

Como resultado, é observado que o dispositivo 5 teve o índice de precisão inalterado, contudo, a revocação foi afetada, o que mostra que várias de suas entradas foram clas- sificadas incorretamente. Olhando a matriz de confusão, é visto que boa parte desses erros ocorreram ao classificar entradas do dispositivo 5 como sendo da classe 6, o que pode indicar a razão do aumento de falsos positivos na

classe 6. Em suma, a retirada da feature de IP do destino resultou na piora da acurácia global em mais de 10%, olhando a acurácia de cada variável pode-se concluir que a classificação se tornou não confiável para pelo menos metade das classes se compararmos com os resultados do modelo L1.

Avaliando as mudanças nas importâncias de cada feature através da Figura 3.16 após a retirada da variável de IP do destino, nota-se que IN_BYTES, que antes era a quarta mais importante, agora se tornou a primeira, e L4_SRC e L4_DST continuaram sendo

Classification Report ----------

	precision	recall	f1-score	support
0	0.00	0.00	0.00	2505
1	0.00	0.00	0.00	3156
2	0.00	0.00	0.00	946
4	1.00	1.00	1.00	19520
5	0.91	0.87	0.89	12490
6	0.00	0.00	0.00	2429
7	1.00	1.00	1.00	62867
8	1.00	0.34	0.51	14832
9	0.37	1.00	0.54	30619
10	1.00	1.00	1.00	90709
11	1.00	0.29	0.45	46054
accuracy			0.81	286127
macro avg	0.57	0.50	0.49	286127
weighted avg	0.90	0.81	0.80	286127

Figura 3.12: Relatório de classificação - Modelo F1.

Confusion Matrix ----------

```
[[    0     0     0     0     1     0     0     0  2486     2    16]
 [    0     0     0     0     2     0     0     0  3154     0     0]
 [    0     0     0     0   107     0     0     0   834     5     0]
 [    0     0     0 19515     0     0     0     0     5     0     0]
 [    0     0     0     0 10843     0     0     5  1642     0     0]
 [    0     0     0     0   954     0     0     0  1475     0     0]
 [    0     0     0     0     0     0 62867     0     0     0     0]
 [    0     0     0     0     7     0     0  5036  9789     0     0]
 [    0     0     0     0    16     0     0     0 30603     0     0]
 [    0     0     0     0     0     0     0     0     1 90708     0]
 [    0     0     0     0     0     0     0     0 32711     0 13343]]
```

Figura 3.13: Matriz de Confusão - Modelo F1.

Classification Report ----------

	precision	recall	f1-score	support
0.0	0.00	0.00	0.00	2505
1.0	1.00	0.97	0.98	3156
2.0	0.00	0.00	0.00	946
3.0	0.00	0.00	0.00	0
4.0	0.97	1.00	0.98	19520
5.0	0.81	0.54	0.65	12490
6.0	0.12	0.37	0.18	2429
7.0	0.73	0.66	0.69	62867
8.0	0.30	0.34	0.32	14832
9.0	0.57	0.73	0.64	30619
10.0	0.92	1.00	0.96	90709
11.0	1.00	0.77	0.87	46054
accuracy			0.79	286127
macro avg	0.53	0.53	0.52	286127
weighted avg	0.80	0.79	0.79	286127

Figura 3.14: Relatório de classificação - L2.

```
Confusion Matrix ----------
[[    0     0     0     0     0     0   301  2201     0     0     0     3]
 [    0  3049     0     6    36    49     6     9     0     0     1     0]
 [    0     0     0    33   345   340   165    39    17     5     2     0]
 [    0     0     0     0     0     0     0     0     0     0     0     0]
 [    0     0     0     0 19518     0     0     0     0     0     0     2]
 [    0     0     0     0   257  6730  4188  1261    50     3     0     1]
 [    0     0     0     0    38   916   904   346   210    14     1     0]
 [    0     0     0     0     0     0     0 41468  3278 14376  3745     0]
 [    0     0     0     0     1    16  1191  8549  5040     3     8    24]
 [    0     0     0     0    10   219   821  2367  4708 22493     0     1]
 [    0     0     0     0     0     0     2     2    50   358 90297     0]
 [    0     0     0     0     0     0     4   295  3505  2430  4230 35590]]
```

Figura 3.15: Matriz de confusão - L2.

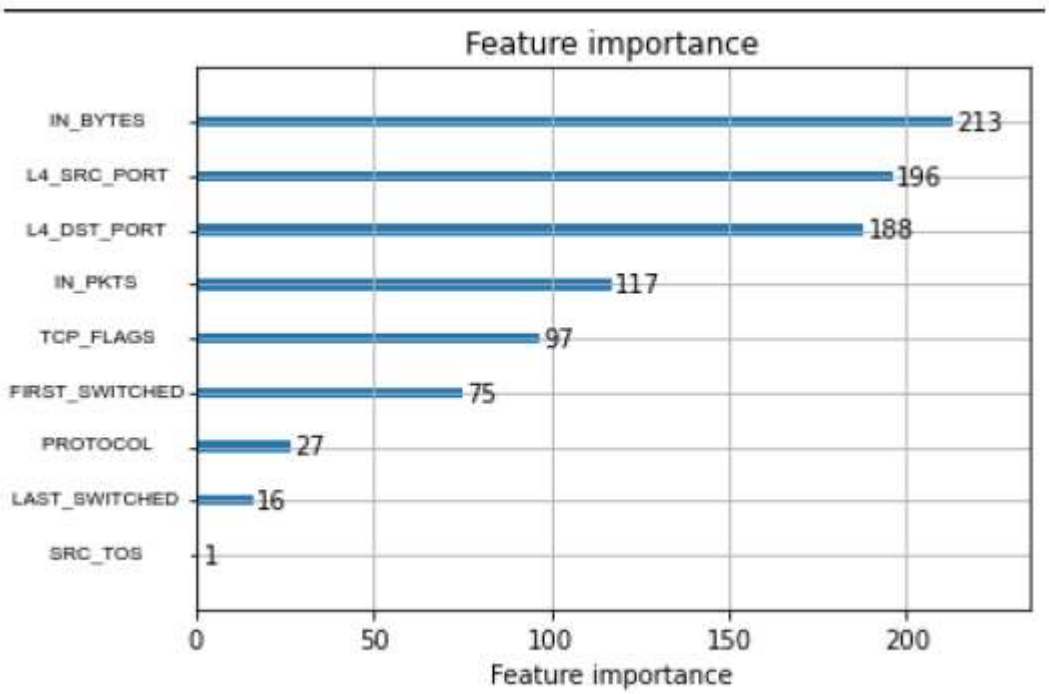


Figura 3.16: Importância das features 2 - LGBM.

terceira e quarta, respectivamente. O aumento da importância da feature IN_PKTS mostra que sem o IP do destino, o número de pacotes de entrada se torna mais importante na classificação.

3.3.2 MODELOS NDF

Com a retirada de IP do destino no treinamento do modelo com o algoritmo NDF hou- veram grandes mudanças, elas foram analisadas a partir da comparação do relatório de classificação, matriz de confusão e curvas de aprendizado dos modelos desse experi- mento com o experimento 1, que gerou os modelos A1 e F1.

MODELO A2

É possível observar através da Figura 3.17 que houve um comportamento diferente durante o treinamento. A acurácia demorou mais pra subir e ficou mais distante de 1, che- gando a 56,88% ao final do treinamento. Já a curva da loss, permaneceu muito próxima da curva do experimento 1, mantendo os valores de máximo e mínimo.

Na fase de testes, os resultados também foram bem diferentes, conforme aponta a figura 3.20, apenas uma classe (dispositivo 10) mostrou resultado satisfatório, com f1- score acima de 90%. Com destaque para a classe 9, que manteve a revocação acima de 90%, mas a precisão reduziu drasticamente pois

nessa classe se concentraram os resultados de classificações incorretas feitas pelo modelo A2, como se vê na Figura 3.21. Desta forma, a acurácia geral do modelo A2 caiu drasticamente.

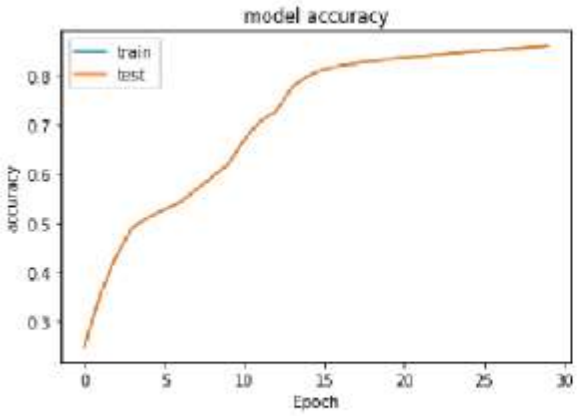


Figura 3.17: Acurácia

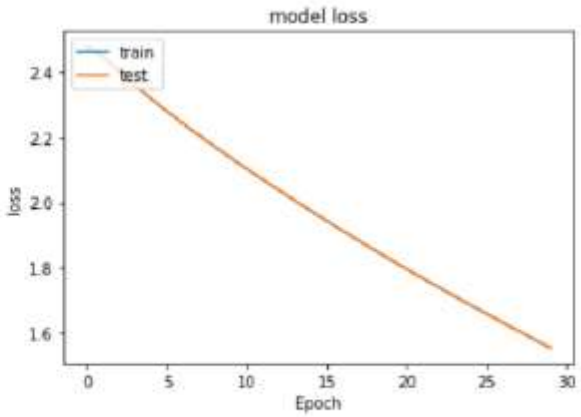


Figura 3.18: Loss

Figura 3.19: Gráfico de aprendizado - Modelo A2.

MODELO F2

Na fase de treinamento do modelo F2 sem a feature de IP do destino, como visto na Figura 3.24 a curva de acurácia é diferente, com aumentos crescentes e sem estabilização até o fim das épocas de treino, talvez, com mais épocas haveriam melhoras, contudo, a tendência é de overfitting.

Na fase de testes, analisando o relatório de classificação através da figura 3.25 e a matriz de confusão na figura 3.26, se observa que os resultados pra todas as classes registraram grande piora e apenas a classe 10 ainda possui índice de acerto aceitável. A matriz de confusão deixa clara a dispersão dos dados, que se aglutinaram especialmente na classe 9, que curiosamente tem índice de revocação alto.

3.4 DISCUSSÃO DE RESULTADOS

A Tabela 3.1 mostra um resumo dos resultados de f1-score de todos os modelos para cada uma das classes, além disso, mostra os resultados de micro avg e acurácia. A tabela facilita a interpretação e a discussão sobre o melhor modelo a ser adotado, destacando os valores mais altos para cada classe. A fim de manter a imparcialidade, no caso de valores de f1-score iguais entre modelos, ambos foram destacados. Os modelos do expe- rimento 1, isto é, L1, A1 e F1, apresentam em geral, melhores resultados. Por isso vamos compará-los entre si, inicialmente.

A maior acurácia é a do modelo L1, e para seis classes, esse modelo obteve o f1- score mais alto. Os melhores valores de f1-score para outras classes em outros modelos, porém, ainda são muito

próximos dos valores no modelo L1. A exceção é para a classe 0, que possui resultado diferente de zero e considerável apenas no modelo A1.

Nos modelos treinados com o experimento 2, que não possuem a variável IPV4, o melhor modelo é L2, também treinado com LGBM. Apesar de resultados mais modestos, ainda possui sete dos melhores resultados para as classes. Contudo, vendo as quedas bruscas nas taxas de acerto para esses modelos, nota-se que a falta da variável de IP do destino realmente afeta consideravelmente os resultados de classificação. Com esta

Classification Report ----------

	precision	recall	f1-score	support
0	0.00	0.00	0.00	2505
1	0.00	0.00	0.00	3156
2	0.00	0.00	0.00	946
3	0.00	0.00	0.00	0
4	0.81	0.01	0.01	19520
5	0.10	0.06	0.08	12490
6	0.00	0.00	0.00	2429
7	0.73	0.65	0.69	62867
8	0.56	0.01	0.01	14832
9	0.23	0.98	0.37	30619
10	1.00	0.91	0.95	90709
11	0.98	0.19	0.32	46054
accuracy			0.57	286127
macro avg	0.37	0.23	0.20	286127
weighted avg	0.75	0.57	0.55	286127

Figura 3.20: Relatório de classificação - Modelo A2.

Confusion Matrix ----------

```
[[    0     0     0     0     0    21     0     1    17  2466     0     0]
 [    0     0     0     0     1   351     0     4     0  2797     1     2]
 [    0     0     0     0     0   226     0     3     0   713     3     1]
 [    0     0     0     0     0     0     0     0     0     0     0     0]
 [    0     0     0     0   143   261     0     0     5 19054    55     2]
 [    0     0     0     1     0   759     0    46     9 11668     7     0]
 [    0     0     0     0     4   127     0    19     0  2273     4     2]
 [    0     0     0     0     6     8     0 40559     0 22294     0     0]
 [    0     0     0     0     0   119     0    22    86 14605     0     0]
 [    0     0     0     0     4   398     0    20    12 29888   120   177]
 [    0     0     0     0    19   325     0    12    11  7785 82554     3]
 [    0     0     0     0     0  4818     0 14606    14 17803    61  8752]]
```

Figura 3.21: Matriz de confusão - Modelo A2.

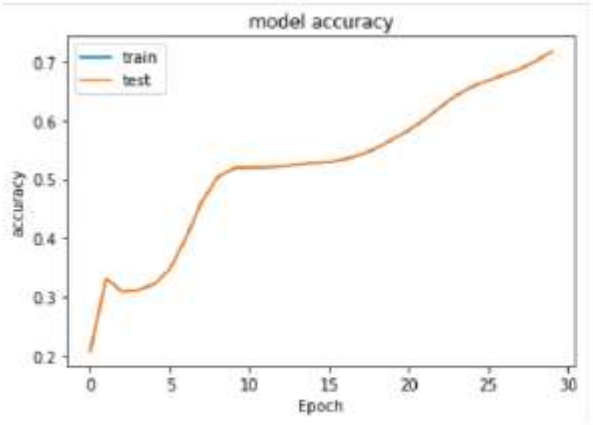


Figura 3.22: Acurácia

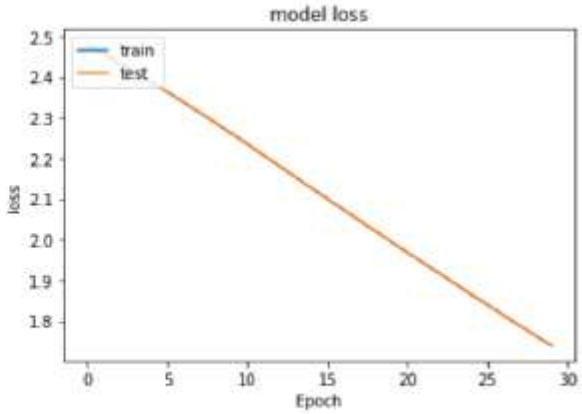


Figura 3.23: Loss

Figura 3.24: Gráfico de aprendizado - Modelo F2.

variável, temos 9 de 12 resultados com taxa de acerto acima de 80%, por outro lado, quando é removida essa variável, restam apenas 5 de 12 classes com taxa de acerto acima de 80%, logo, para mais da metade das classes, a classificação é ineficaz.

Classification Report ----------

	precision	recall	f1-score	support
0	0.00	0.00	0.00	2505
1	0.00	0.00	0.00	3156
2	0.00	0.00	0.00	946
4	0.00	0.00	0.00	19520
5	0.18	0.00	0.00	12490
6	0.00	0.00	0.00	2429
7	0.67	1.00	0.80	62867
8	1.00	0.34	0.51	14832
9	0.40	0.95	0.56	30619
10	0.92	1.00	0.95	90709
11	1.00	0.32	0.49	46054
accuracy			0.71	286127
macro avg	0.38	0.33	0.30	286127
weighted avg	0.70	0.71	0.64	286127

Figura 3.25: Relatório de classificação 2 - Modelo F2.

```
Confusion Matrix ----------------------
[[    0     0     0     0     0     0     3     0  1504   998     0]
 [    0     0     0     0     2     0     1     0  2793   353     7]
 [    0     0     0     0     0     0     2     0   692   251     1]
 [    0     0     0     0     0     0    26     0 19142   324    28]
 [    0     0     0     0    30     0     4     5 11478   972     1]
 [    0     0     0     0     7     0     4     0  2268   144     6]
 [    0     0     0     0     0     0 62865     0     0     0     2]
 [    0     0     0     0     1     0    27  5036  5745  4011    12]
 [    0     0     0     0   128     0   192     2 28988  1304     5]
 [    0     0     0     0     0     0     8     0   409 90291     1]
 [    0     0     0     0     0     0 31114     0    73    10 14857]]
```

Figura 3.26: Matriz de Confusão 2 - Modelo F2.

Tabela 3.1: Resumo dos resultados.

Resumo dos resultados						
Modelos	L1	A1	F1	L2	A2	F2
Classes	f1-score					
0	0,00	**0,93**	0,00	0,00	0,00	0,00
1	**0,70**	0,47	0,00	**0,98**	0,00	0,00
2	**0,02**	0,00	0,00	0,00	0,00	0,00
3	0,00	0,00	0,00	0,00	0,00	0,00
4	0,98	**1,00**	**1,00**	**0,98**	0,01	0,00
5	0,80	0,54	**0,89**	**0,65**	0,08	0,00
6	**0,20**	0,00	0,00	**0,18**	0,00	0,00
7	0,96	**1,00**	**1,00**	0,69	0,69	**0,80**
8	**0,87**	0,00	0,51	0,32	0,01	**0,51**
9	**0,94**	0,63	0,54	**0,64**	0,37	0,56
10	0,97	**1,00**	**1,00**	**0,96**	0,95	0,95
11	**0,98**	0,47	0,45	**0,87**	0,32	0,49
Macro AVG	**0,62**	0,50	0,49	**0,52**	0,20	0,30
Acurácia	**0,93**	0,79	0,80	**0,79**	0,55	0,64

CAPÍTULO 4

CONCLUSÃO E TRABALHOS FUTUROS

Neste trabalho, foi avaliado o desempenho e comparadas as métricas de dois mode- los treinados com base em um dataset real coletado a partir de dispositivos IoT e não IoT conectados a uma rede. Alguns parâmetros principais foram definidos para que a compa- ração entre os modelos pudesse ser feita de maneira justa, e dois algoritmos diferentes foram escolhidos para gerarem os modelos. Assim, foi possível comparar a acurácia de ambos ao classificar os dispositivos que perdem sua identificação quando passam pela NAT no tráfego da rede doméstica para a telco.

Foi levantada a problemática da presença do atributo IPV4 nos dados do dataset, e a possibilidade do enviesamento. Com base nos gráficos e tabelas comparativas dos resultados dos modelos, é notável que apesar de bons acertos, alguns dispositivos não são classificados corretamente de maneira satisfatória, e que o algoritmo baseado em redes neurais, não parece ser a melhor abordagem para este tipo de problema.

Há tópicos a serem explorados como trabalhos futuros: a) avaliação de outros algo- ritmos de classificação buscando aumentar a acurácia em dispositivos com alta taxa de classificações incorretas e b) configuração de randomização em rotinas de requisição de dispositivos IoT com objetivo de dificultar, ou, impossibilitar a classificação.

REFERÊNCIAS BIBLIOGRÁFICAS

Hummel, Richard, Carol Hildebrand, Hardik Modi, Chris Conrad, Roland Dobbins, Steinthor Bjarnson, Jon Belanger, Gary Sockrider & Tom Bienkowski Philippe Alcoy (2020), 'Netscout threat intelligence report issue 6: Findings from 2h 2020 ddos in a time of pandemic including the 16th annual worldwide infrastructure security report (wisr) if you are looking for real-time data on global ddos attacks, cyber threat horizon is an invaluable (and free) tool'.

Kontschieder, Peter, Madalina Fiterau & Samuel Rota Bulo Antonio Criminisi (2015), 'Deep neural decision forests', 2015 IEEE International Conference on Computer Vision (ICCV) pp. 1467–1475.

URL: http://ieeexplore.ieee.org/document/7410529/

Lipton, Zachary Chase, Charles Elkan & Balakrishnan Narayanaswamy (2014), 'Threshol- ding classifiers to maximize f1 score', arXiv preprint arXiv:1402.1892 .

Meidan, Yair, Vinay Sachidananda, Hongyi Peng, Racheli Sagron, Yuval Elovici & Asaf Shabtai (2020), 'A novel approach for detecting vulnerable iot devices connected behind a home nat', Computers and Security 97.

Meneghello, Francesca, Matteo Calore, Daniel Zucchetto, Michele Polese & Andrea Za- nella (2019), 'Iot: Internet of threats? a survey of practical security vulnerabilities in real iot devices', IEEE Internet of Things Journal 6, 8182–8201.

Wang, An, Wentao Chang, Songqing Chen & Aziz Mohaisen (2018), 'Delving into internet ddos attacks by botnets: Characterization and analysis', IEEE/ACM Transactions on Networking 26, 2843–2855.

Capítulo 6

10.37423/220706251

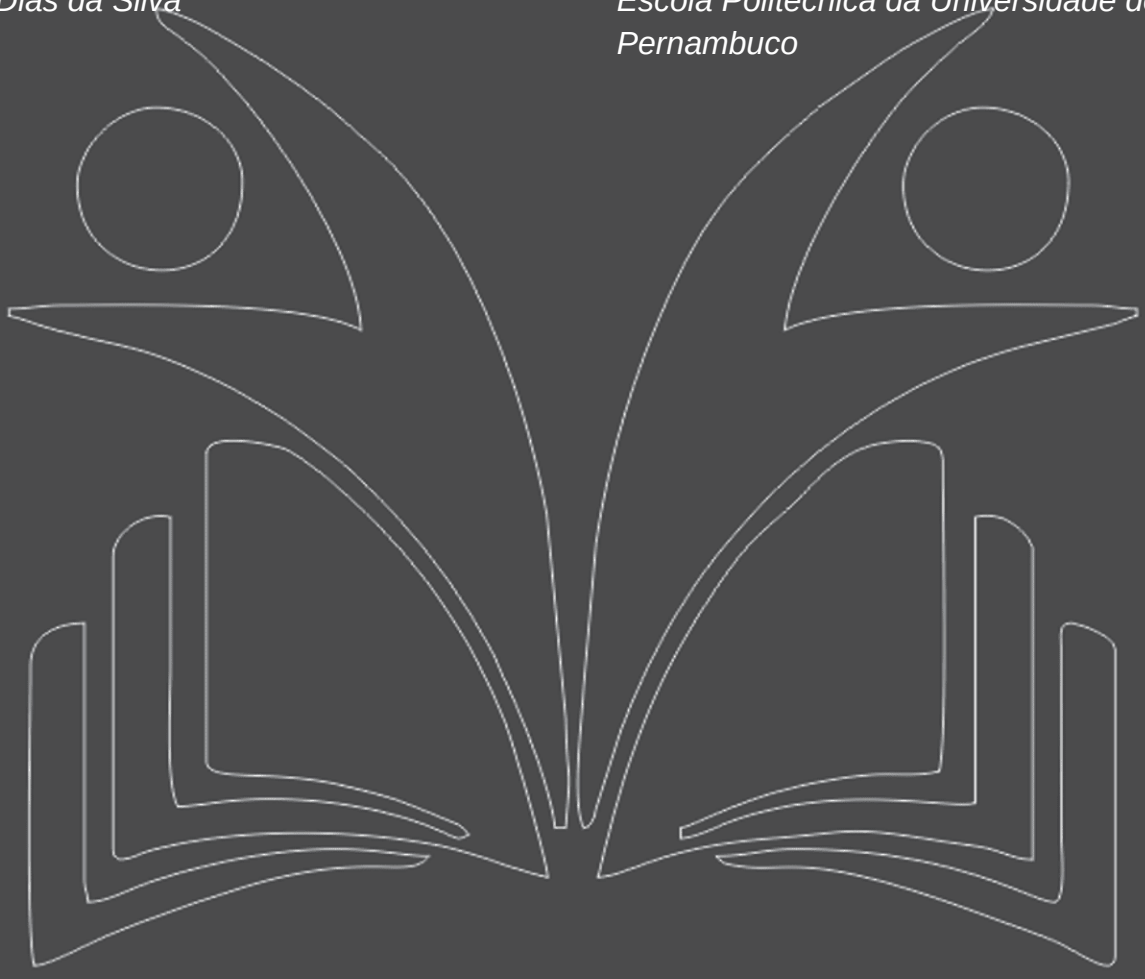
INTERVAL-VALUED DATA MEDIAN CLUSTERING (IWPGMC): STEP-BY-STEP MIDPOINT UPDATE FORMULA

Sérgio Mário Lins Galdino

Escola Politécnica da Universidade de Pernambuco

Jornandes Dias da Silva

Escola Politécnica da Universidade de Pernambuco



Abstract: The weighted centroid clustering (WPGMC) is one popular hierarchical cluster algorithm (HCA). Currently, clustering methods depends on dissimilarity measures for interval- valued data uses representative point distance. IWPGMC extends Group Average clustering to interval-valued data. Based on the Range Euclidean Metric it is a reliable alternative to be used to uncertainty quantification from interval-valued data. They contain more information than point-valued data, and such informational advantages could be exploited to yield more efficient analysis.

Keywords: interval-valued data, hierarchical clustering, interval analysis.

I. INTRODUCTION

In cluster analysis, a huge number of methods are available for classifying objects on the basis of their (dis)similarities. Major types of cluster analysis are hierarchical methods (agglomerative or divisive), partitioning methods, and methods that allow overlapping clusters [1].

Hierarchical clustering cluster analysis (HCA), is a whole family of algorithms that differ by distance updating. The seven popular methods include Single Linkage, Complete Linkage, Simple Average (WPGMA -Weighted Pair Group Method Average), Group Average (UPGMA -Unweighted Pair Group Method Average), Median (WPGMC -Weighted Pair Group Method Centroid), Centroid (UPGMC -Unweighted Pair Group Method Centroid), and Ward's Minimum Variance Method. They are implemented in standard numerical and statistical software such as Octave, Matlab, SciPy, Mathematica, R.

The goal of hierarchical cluster analysis is to build a tree diagram where the objects that were viewed as most similar by the participants in the study are placed on branches that are close together. The key to interpreting a hierarchical cluster analysis is to look at the point at which any given pair of objects "join together" in the tree diagram. Objects that join sooner are more similar to each other than those that join later. Median, or equilibrious centroid method (WPGMC) is the modified Centroid method (UPGMC). The WPGMC differs from UPGMC by weighting the member most recently admitted to a cluster equal with all previous members. Proximity between two clusters is the proximity between their geometric centroids ([squared] Euclidean distance between those); while the centroids are defined so that the subclusters of which each of these two clusters were merged recently have equalized influence on its centroid – even if the subclusters differed in the number of objects. Name "median" is partly misleading because the method doesn't use medians of data distributions, it is still based on centroids.

Traditional clustering techniques can be easily applied to interval data types by replacing each interval with a representative (e.g., the median of the points in the interval). However, this approach ignores the structure information of the interval. For example, A= [2, 4], B= [1, 5] and C= [7,9] are three interval-valued data objects. A and B have the same centroids. The representative centroid of C has the same Euclidean distance to the representative centroids of both A and B. Thus, we have limitations of using representative centroids to replace intervals.

Granular data and models offer an interesting tool for rep- resenting data in problems involving uncertainty, inaccuracy, variability and subjectivity have to be taken into account. Interval analysis is fundamental of supporting computation of interval granular fuzzy model [2], [3].

In a seminal paper, Se´rgio Galdino [4] used Interval-valued Data Hierarchical Cluster Analysis - IHCA, clustering based on single-linkage clustering method. The Range City Block Metric for interval-valued data is used to compare two vectors of intervals. Se´rgio Galdino & Paulo Maciel [5] presents the Simple Average (IWPGMA) clustering for interval-valued data. Se´rgio Galdino & Jornandes Dias introduces Interval-valued Data Ward's Minimum Variance Clustering using Centroid update formula [6].

In this paper, we deal with a particular type of information granules, namely interval-valued data. We face the problem of clustering data units described by variables whose values are intervals of the real data set (interval data). We will focus on IWPGMC (Interval-valued Data Weighted Pair-Group Method with Centroid). The Range Euclidean Metric for interval- valued data is used to compare two vectors of intervals.

The rest of the paper is organized as follows: Section II presents basic concepts of interval arithmetic and distance measures for interval data. Section III describes interval-valued input distance matrix for clustering. Section IV introduces the approach for clustering interval data, Section V provides the conclusion and discusses future work.

II. INTERVAL ARITHMETIC

Interval arithmetic is a method for determining absolute errors of an algorithm, considering all data errors and rounding [7]. Interval arithmetic makes systematic calculations through intervals $[x] = \left[\underline{x}, \overline{x}\right]$ limited to representable machine numbers $\underline{x}, \overline{x} \in F$, instead of real numbers x. Arithmetic operations $+$, $-$, $\times$, $\div$ are defined using intervals. Interval algorithms produce interval results guaranteed to contain the true solution. If $\bullet$ denotes any of these arithmetic operation for real numbers x and $y$then the corresponding operation for arithmetic on interval numbers $[x]$ and $[\mathrm{y}]$ is

$$[\mathrm{x}] \bullet [\mathrm{y}] = \{\mathrm{x} \bullet \mathrm{y} | \mathrm{x} \in [\mathrm{x}], \mathrm{y} \in [\mathrm{y}]\}.$$

Thus, the interval $[\mathrm{x}] \bullet [\mathrm{y}]$ resulting from the operations contain every possible number that can be found as $\mathrm{x} \bullet \mathrm{y}$ for each $\mathrm{x} \in [\mathrm{x}]$, and each $\mathrm{y} \in [\mathrm{y}]$.

The interval arithmetic operations are defined for exact calculation [7]. Machine computations are affected by rounding errors. Therefore, the formulas were modified in order to consider the called directed rounding [8].

Throughout this paper, all matrices are denoted by bold capital letters ($\boldsymbol{A}$), vectors by bold lowercase letters ($\boldsymbol{a}$)and scalar variables by ordinary lowercase letters ($\mathbf{a}$). Interval objects are enclosed in square brackets($[\boldsymbol{A}], [\boldsymbol{a}], [\mathrm{a}]$). Underscores and overscores denote lower and upper bounds, respectively.

Definition. Given two intervals$[x], [y] \in I(R)$, $[x] \leq [y], iff\ m([x]) \leq m([y])$, where $m(\mathcal{X})$ is a point within the interval $\mathcal{X} \in \{ [x], [y] \}$, usually the *midpoint*, *infimum*, and *supremum*. We propose the following order relation: $[x] \leq [y]$ is determined by choosing the interval *infimum* that captures the "minimum" between the two intervals, *i.e.,* the interval with the lowest *infimum.*

A. RANGE OF INTERVAL-VALUED FUNCTION

The range of an interval-valued function can be expressed in interval form as $\mathrm{range}\big(\mathrm{f}([\boldsymbol{x}])\big) = \mathrm{f}([\mathrm{x}_1], [\mathrm{x}_2], \dots, [\mathrm{x}_n])$

$$= [inf f([x_1], \dots, [x_n]), sup f([x_1], \dots, [x_n])] \qquad (1)$$

where the *inf* and *sup* are taken for all $x_i \in [x_i] (i = 1, \dots n)$.

Finding the range of a multi-variable function over a box is a fundamental problem encountered in numerous applications. The main focus of interval arithmetic is the simplest way to calculate upper and lower endpoints for the range of values of a function in one or more variables. These endpoints are not necessarily the *supremum* or *infimum*, since the precise calculation of those values can be difficult or impossible. In special cases the exact range can be found in a straightforward way [9], [10].

B. RANGE EUCLIDEAN DISTANCE

The Range Euclidean Distance between interval vectors $[p]$ and $[q]$ is the interval length of the lines segment connecting them $\overline{[\mathbf{p}][\mathbf{q}]}$.

In cartesian coordinates, if $[\mathbf{p}] = ([p_1], [p_2], \dots, [p_n])$ and $[\mathbf{q}] = ([q_1], [q_2], \dots, [q_n])$ are two interval vectors in Euclidean n-space (i.e., $I(R^n)$), then the distance $[d]$ from $[\boldsymbol{p}]$ to $[\boldsymbol{q}]$, or from $[\boldsymbol{q}]$ to $[\boldsymbol{p}]$ is given by the Interval Pythagorean formula:

$$d_2([\boldsymbol{p}], [\boldsymbol{q}]) = d_2([\boldsymbol{q}], [\boldsymbol{p}]) =$$

$$= \sqrt{([q_1] - [p_1])^2 + ([q_2] - [p_2])^2 + \cdots + ([q_n] - [p_n])^2}$$

$$= \sqrt{\sum_{i=1}^{n} ([q_i] - [p_i])^2} \qquad (2)$$

Consider the function x^2 as a monotonically decreasing function for $x < 0$ and a monotonically increasing function for $x > 0$. Interval function can be defined. The range corresponding to the interval $[x]^2$ can be calculated by applying the function to its endpoints:

$$[x_1, x_2]^2 = \begin{cases} [x_1^2, x_2^2] & x_1 \geq 0 \\ [x_2^2, x_1^2] & x_2 < 0 \\ 0, max\{x_1^2,\ x_2^2\} & otherwise \end{cases} \qquad (3)$$

III. INTERVAL-VALUED INPUT DISTANCE MATRIX

Table I provides the oils dataset originally presented by

Ichino [11].

TABLE I: interval-valued observations: oils and fats.

Sample (Label)	Specific gravity	Freezing point	Iodine value	Saponification value
Linseed oil (A)	[0.930, 0.935]	[−27, −18]	[170, 204]	[118, 196]
Perilla oil (B)	[0.930, 0.937]	[−5, −4]	[192, 208]	[188, 197]
Cottonseed oil (C)	[0.916, 0.918]	[−6, −1]	[99, 113]	[189, 198]
Sesame oil (D)	[0.92, 0.926]	[−6, −4]	[104, 116]	[187, 193]
Camellia oil (E)	[0.916, 0.917]	[−21, −15]	[80, 82]	[189, 193]
Olive oil (F)	[0.914, 0.919]	[0, 6]	[79, 90]	[187, 196]
Beef tallow (G)	[0.86, 0.87]	[30, 38]	[40, 48]	[190, 199]
Hog fat (H)	[0.858, 0.864]	[22, 32]	[53, 77]	[190, 202]

https://en.wikipedia.org/wiki/Saponification_value

IV. AGGLOMERATIVE HIERARCHICAL CLUSTERING

Agglomerative clustering works in a “bottom-up” manner. That is, each object is initially considered as a single-element cluster (leaf). At each step of the algorithm, the two clusters that are the most similar are combined into a new bigger cluster (nodes). This procedure is iterated until all points are member of just one single big cluster (root) (see below).

Hierarchical Clustering Algorithms

Given a set of N objects (i.e., observations, individuals, cases, or data rows) to be clustered, and an NxN distance (or similarity) matrix, the basic process of Johnson's [12] hierarchical clustering is this:

1) Start by assigning each object to its own cluster, so that if you have N objects, you now have N clusters, each containing just one item. Let the distances (similarities) between the clusters equal the distances (similarities) between the objects they contain.
2) Find the closest (most similar) pair of clusters and merge them into a single cluster, so that now you have one less cluster.
3) Compute distances (similarities) between the new cluster and each of the old clusters.
4) Repeat steps 2 and 3 until all items are clustered into a single cluster of size N.

Step 3 can be done in different ways, which is what distinguishes HCA (Hierarchical Clustering Algorithms). Should be stressed that in Step 2, we will use *infimum* to find the closest pair of clusters. Should be stressed that in Step 2, we will use *infimum* of interval to find the closest pair of clusters. The range metrics gives make possible others merge points (see Section II).

A. IWPGMC - MIDPOINT UPDATE FORMULA

The IWPGMC (*I*nterval-valued *D*ata *W*eighted *P*air-*G*roup *M*ethod with *C*entroid) - midpoint update formula algorithm produces rooted dendrograms. At each step, the nearest two clusters, say $[A]$ and $[B]$, are combined into a higher-level cluster $[A] \cup [B]$.

Applying the interval-valued data IWPGMC - midpoint update formula method we merge $[A]$ and $[B]$. That is exactly where the linkage rule comes into action. Using IUPGMC cluster dissimilarity between clusters $[X]$ and $[Y]$ update formula:

$$d([X],[Y]) = \left|\left|\overrightarrow{[w]}_{[X]} - \overrightarrow{[w]}_{[Y]}\right|\right|_2 \qquad (4)$$

The interval midpoint is defined iteratively and depends on the clustering steps:

$$\overrightarrow{[w]}_{[C]} = \frac{1}{2}\left(\overrightarrow{[w]}_{[A]} + \overrightarrow{[w]}_{[B]}\right) \qquad (5)$$

denotes the interval midpoint of a cluster $[C] = \{[X],[Y]\}$.

FIRST STEP

First clustering: Let us assume that we have eight elements (A, B, C, D, E, F, G) with interval midpoints Table II and initial distance matriz Table III (using Equation (4)). From distance matrix Table III of pairwise distances between them: $d_{(C,D)} = [0.002,\ 20.857]$ has the smallest *infimum* value of distance matrix, so we join elements C and D.

TABLE II: Interval midpoints of Clusters.

$\overrightarrow{[w]}_{[A]} =$	[0.930, 0.935]	[-27, -18]	[170, 204]	[118, 196]
$\overrightarrow{[w]}_{[B]} =$	[0.930, 0.937]	[-5, -4]	[192, 208]	[188, 197]
$\overrightarrow{[w]}_{[C]} =$	[0.916, 0.918]	[-6, -1]	[99, 113]	[189, 198]
$\overrightarrow{[w]}_{[D]} =$	[0.92, 0.926]	[-6, -4]	[104, 116]	[187, 193]
$\overrightarrow{[w]}_{[E]} =$	[0.916, 0.917]	[-21, -15]	[80, 82]	[189, 193]
$\overrightarrow{[w]}_{[F]} =$	[0.914, 0.919]	[0, 6]	[79, 90]	[187, 196]
$\overrightarrow{[w]}_{[G]} =$	[0.86, 0.87]	[30, 38]	[40, 48]	[190, 199]
$\overrightarrow{[w]}_{[H]} =$	[0.858, 0.864]	[22, 32]	[53, 77]	[190, 202]

TABLE III: Lower Non-Diagonal Distance matrix.

	A	B	C	D	E	F	G
B	[13, 90.632]						
C	[58.249, 134.54]	[79, 109.54]					
D	[55.317, 127.1]	[76, 104.5]	[0.002, 20.857]				
E	[88, 145.42]	[110.45, 129.38]	[19.235, 39.624]	[23.769, 40.262]			
F	[82, 151]	[102.07, 129.86]	[9.0553, 37.697]	[14.56, 39.925]	[15, 29.632]		
G	[131.1, 194.12]	[147.95, 173.77]	[59.682, 85.82]	[65.513, 88.635]	[55.217, 73.11]	[39.204, 63.938]	
H	[101.23, 182.59]	[117.9, 159.97]	[31.827, 72.202]	[37.483, 75.087]	[37.121, 61.799]	[16.124, 51.167]	[5, 42.06]

Min Distance (Linkage)

First midpoint update:

$$\overrightarrow{[w]}_{[(C,D)]} = \frac{1}{2}\left(\overrightarrow{[w]}_{[C]} + \overrightarrow{[w]}_{[D]}\right)$$

We then proceed to update the initial distance matrix Table II into a new distance matrix Table IV, reduced in size by one row and one column because of the clustering of C with D. Using Table IV, update distances between clusters are computed as:

$$\begin{aligned}
d_{(C,D)\to A} &= \left\| \overrightarrow{[w]}_{[(C,D)]} - \overrightarrow{[w]}_{[A]} \right\|_2 = [56.782,\ 130.82] \\
d_{(C,D)\to B} &= \left\| \overrightarrow{[w]}_{[(C,D)]} - \overrightarrow{[w]}_{[B]} \right\|_2 = [77.5,\ 106.91] \\
d_{(C,D)\to E} &= \left\| \overrightarrow{[w]}_{[(C,D)]} - \overrightarrow{[w]}_{[E]} \right\|_2 = [21.476,\ 39.684] \\
d_{(C,D)\to F} &= \left\| \overrightarrow{[w]}_{[(C,D)]} - \overrightarrow{[w]}_{[F]} \right\|_2 = [11.768,\ 38.426] \\
d_{(C,D)\to G} &= \left\| \overrightarrow{[w]}_{[(C,D)]} - \overrightarrow{[w]}_{[G]} \right\|_2 = [62.597,\ 87.22] \\
d_{(C,D)\to H} &= \left\| \overrightarrow{[w]}_{[(C,D)]} - \overrightarrow{[w]}_{[H]} \right\|_2 = [34.648,\ 73.636]
\end{aligned}$$

SECOND STEP

Second clustering: We now reiterate the three previous steps starting from the new distance matrix Table V. Here, $d_{(G,H)} = [5,\ 42.06]$ has the lowest *infimum* value of Table V, so we join elements G and H.

Second midpoint update:

$$\overrightarrow{[w]}_{[(G,H)]} = \frac{1}{2}\left(\overrightarrow{[w]}_{[G]} + \overrightarrow{[w]}_{[H]}\right)$$

TABLE IV: Interval midpoints of Clusters - Grouped Cluster (C, D).

$\overrightarrow{[w]}_{[A]} =$	[0.930, 0.935]	[-27, -18]	[170, 204]	[118, 196]
$\overrightarrow{[w]}_{[B]} =$	[0.930, 0.937]	[-5, -4]	[192, 208]	[188, 197]
$\overrightarrow{[w]}_{[(C,D)]} =$	[0.918, 0.922]	[-6, -2.5]	[101.5,, 114.5]	[188, 195.5]
$\overrightarrow{[w]}_{[E]} =$	[0.916, 0.917]	[-21, -15]	[80, 82]	[189, 193]
$\overrightarrow{[w]}_{[F]} =$	[0.914, 0.919]	[0, 6]	[79, 90]	[187, 196]
$\overrightarrow{[w]}_{[G]} =$	[0.86, 0.87]	[30, 38]	[40, 48]	[190, 199]
$\overrightarrow{[w]}_{[H]} =$	[0.858, 0.864]	[22, 32]	[53, 77]	[190, 202]

TABLE V: Grouped Cluster (C, D).

	A	B	C, D	E	F	G
B	[13, 90.632]					
C, D	**[56.782, 130.82]**	**[77.5, 106.91]**				
E	[88, 145.42]	[110.45, 129.38]	**[21.476, 39.684]**			
F	[82, 151]	[102.07, 129.86]	**[11.768, 38.426]**	[15, 29.632]		
G	[131.1, 194.12]	[147.95, 173.77]	**[62.597, 87.22**	[55.217, 73.11]	[39.204, 63.938]	
H	[101.23, 182.59]	[117.9, 159.97]	**[34.648, 73.636]**	[37.121, 61.799]	[16.124, 51.167]	[5, 42.06]

Min Distance (Linkage)

We then proceed to update midpoints Table IV into a new midpoints Table VI, reduced in size by one row because of the clustering of G with H.

Second distance matrix update: We then proceed to update the matrix Table V into a new distance matrix Table VII, reduced in size by one row and one column because of the clustering of *G* with H. Using Table VI, update distances between clusters are computed as:

$$d_{(G,H)\to A} = \left|\left|\overrightarrow{[w]}_{[(G,H)]} - \overrightarrow{[w]}_{[A]}\right|\right|_2 = [116.15,\ 188.3]$$
$$d_{(G,H)\to B} = \left|\left|\overrightarrow{[w]}_{[(G,H)]} - \overrightarrow{[w]}_{[B]}\right|\right|_2 = [132.92,\ 166.85]$$
$$d_{(G,H)\to (C,D)} = \left|\left|\overrightarrow{[w]}_{[(G,H)]} - \overrightarrow{[w]}_{[(C,D)]}\right|\right|_2 = [48.303,\ 80.382]$$
$$d_{(G,H)\to E} = \left|\left|\overrightarrow{[w]}_{[(G,H)]} - \overrightarrow{[w]}_{[E]}\right|\right|_2 = [44.578,\ 67.295]$$
$$d_{(G,H)\to F} = \left|\left|\overrightarrow{[w]}_{[(G,H)]} - \overrightarrow{[w]}_{[F]}\right|\right|_2 = [25.927, 57.442]$$

TABLE VI: Interval midpoints of Clusters - Grouped Cluster (G, H).

$\overrightarrow{[w]}_{[A]}$ =	[0.930, 0.935]	[-27, -18]	[170, 204]	[118, 196]
$\overrightarrow{[w]}_{[B]}$ =	[0.930, 0.937]	[-5, -4]	[192, 208]	[188, 197]
$\overrightarrow{[w]}_{[(C,D)]}$ =	[0.918, 0.922]	[-6, -2.5]	[101.5,, 114.5]	[188, 195.5]
$\overrightarrow{[w]}_{[E]}$ =	[0.916, 0.917]	[-21, -15]	[80, 82]	[189, 193]
$\overrightarrow{[w]}_{[F]}$ =	[0.914, 0.919]	[0, 6]	[79, 90]	[187, 196]
$\overrightarrow{[w]}_{[(G,H)]}$ =	[0.859, 0.867]	[26, 35]	[46.5, 62.5]	[190, 200.5]

TABLE VII: Grouped Cluster (G, H).

	A	B	C, D	E	F
B	[13, 90.632]				
C, D	[56.782, 130.82]	[77.5, 106.91]			
E	[88, 145.42]	[110.45, 129.38]	[21.476, 39.684]		
F	[82, 151]	[102.07, 129.86]	[11.768, 38.426]	[15, 29.632]	
G, H	**[116.15, 188.3]**	**[132.92, 166.85]**	**[48.303, 80.382]**	**[44.578, 67.295]**	**[25.927, 57.442]**

Min Distance (Linkage)

Third step

Third clustering: We now reiterate starting from the new distance matrix Table VII. Here, $d_{((C,D),F)} = [11.768,\ 38.426]$ has the lowest *infimum* value of Table VII, so we join elements (C, D) and F.

Third midpoint update:

$$\overrightarrow{[w]}_{[((C,D),\ F)]} = \frac{1}{2}\left(\overrightarrow{[w]}_{[(C,D)]} + \overrightarrow{[w]}_{[F]}\right)$$

We then proceed to update midpoints Table VI into a new midpoints Table VII, reduced in size by one row because of the clustering of (C, D) with F.

Third distance matrix update: We then proceed to update the matrix Table VII into a new distance matrix Table IX, reduced in size by one row and one column because of the clustering of (C, D) with F. Using Table VIII, update distances between clusters are computed as:

$$d_{(C,D),F)\to A} = \left|\left|\overrightarrow{[w]}_{[((C,D),F)]} - \overrightarrow{[w]}_{[A]}\right|\right|_2 = [69.39,\ 140.76]$$
$$d_{(C,D),F)\to B} = \left|\left|\overrightarrow{[w]}_{[((C,D),F)]} - \overrightarrow{[w]}_{[B]}\right|\right|_2 = [89.755,\ 118.33]$$
$$d_{(C,D),F)\to E} = \left|\left|\overrightarrow{[w]}_{[((C,D),F)]} - \overrightarrow{[w]}_{[E]}\right|\right|_2 = [14.562,\ 32.53]$$
$$d_{(C,D),F)\to (G,H)} = \left|\left|\overrightarrow{[w]}_{[((C,D),F)]} - \overrightarrow{[w]}_{[(G,H)]}\right|\right|_2 = [36.852, 68.7$$

TABLE VIII: Interval midpoints of Clusters - Grouped Cluster ((C, D), F).

$\overrightarrow{[w]}_{[A]} =$	[0.930, 0.935]	[-27, -18]	[170, 204]	[118, 196]
$\overrightarrow{[w]}_{[B]} =$	[0.930, 0.937]	[-5, -4]	[192, 208]	[188, 197]
$\overrightarrow{[w]}_{[((C,D),F)]} =$	[0.916, 0.9205]	[-3, +1.75]	[90.25, 102.25]	[187.5, 195.75]
$\overrightarrow{[w]}_{[E]} =$	[0.916, 0.917]	[-21, -15]	[80, 82]	[189, 193]
$\overrightarrow{[w]}_{[(G,H)]} =$	[0.859, 0.867]	[26, 35]	[46.5, 62.5]	[190, 200.5]

TABLE IX: Grouped Cluster $((C, D), F)$.

	A	B	(C, D), F	E
B	[13, 90.632]			
(C, D), F	**[69.39, 140.76]**	**[89.755, 118.33]**		
E	[88, 145.42]	[110.45, 129.38]	**[14.562, 32.53]**	
G, H	[116.15, 188.3]	[132.92, 166.85]	**[36.852, 68.711]**	[44.578, 67.295]

Min Distance (Linkage)

FOURTH STEP

Fourth clustering: We now reiterate starting from the new distance matrix Table IX. Here, $d_{(A,B)} = [13,\ 90.632]$ has the lowest *infimum* value of Table IX, so we join element A with cluster B.

Fourth midpoint update:

$$\overrightarrow{[w]}_{[(A,B)]} = \frac{1}{2}\left(\overrightarrow{[w]}_{[A]} + \overrightarrow{[w]}_{[B]}\right)$$

We then proceed to update midpoints Table VIII into a new midpoints Table X, reduced in size by one row because of the clustering of A with B.

Fourth distance matrix update: We then proceed to update the matrix Table IX into a new distance matrix Table XI, reduced in size by one row and one column because of the clustering of A with B. Using the Table X, update distances between clusters are computed as:

$$d_{(A,B)\to((C,D),F)} = \left|\left|\overrightarrow{[w]}_{[(A,B)]} - \overrightarrow{[w]}_{[((C,D),F)]}\right|\right|_2 = [79.155,\ 124.67]$$
$$d_{(A,B)\to E} = \left|\left|\overrightarrow{[w]}_{[(A,B)]} - \overrightarrow{[w]}_{[E]}\right|\right|_2 = [99,\ 132.58]$$
$$d_{(A,B)\to(G,H)} = \left|\left|\overrightarrow{[w]}_{[(A,B)]} - \overrightarrow{[w]}_{[(G,H)]}\right|\right|_2 = [124.14,\ 174.07]$$

FIFTH STEP

Fifth clustering: We now reiterate starting from the new distance matrix Table XI. Here, $d_{(((C,D),F),E)} = [14.562,\ 32.53]$ has the lowest *infimum*} value of Table XI, so we join element $(C, D), F$ with cluster E.

Fifth midpoint update:

$$\overrightarrow{[w]}_{[(((C,D),F),E)]} = \frac{1}{2}\left(\overrightarrow{[w]}_{[((C,D),F)]} + \overrightarrow{[w]}_{[E]}\right)$$

We then proceed to update midpoints Table X into a new midpoints Table XII, reduced in size by one row because of the clustering of $(C, D), F$ with E

TABLE X: Interval midpoints of Clusters - Grouped Cluster (A, B).

$\overrightarrow{[w]}_{[(A,B)]}$ =	[0.930, 0.936]	[-16, -11]	[181, 206]	[153, 196.5]
$\overrightarrow{[w]}_{[((C,D),F)]}$ =	[0.916, 0.9205]	[-3, +1.75]	[90.25, 102.25]	[187.5, 195.75]
$\overrightarrow{[w]}_{[E]}$ =	[0.916, 0.917]	[-21, -15]	[80, 82]	[189, 193]
$\overrightarrow{[w]}_{[(G,H)]}$ =	[0.859, 0.867]	[26, 35]	[46.5, 62.5]	[190, 200.5]

TABLE XI: Grouped Cluster (A, B).

	A, B	(C, D), F	E
(C, D), F	**[79.155, 124.67]**		
E	**[99, 132.58]**	[14.562, 32.53]	
G, H	**[124.14, 174.07]**	[36.852, 68.711]	[44.578, 67.295]

Min Distance (Linkage)

Fifth distance matrix update: We then proceed to update the matrix Table XI into a new distance matrix Table XIII, reduced in size by one row and one column because of the clustering of $(C, D), F$ with E. Using the Table XII, update distances between clusters are computed as:

$$\begin{aligned} d_{(((C,D),F),E)\to(A,B)} &= \left|\left|\overrightarrow{[w]}_{[(((C,D),F),E)]} - \overrightarrow{[w]}_{[(A,B)]}\right|\right|_2 \\ &= [88.875,\ 128.11] \\ d_{(((C,D),F),E)\to(G,H)} &= \left|\left|\overrightarrow{[w]}_{[(((C,D),F),E)]} - \overrightarrow{[w]}_{[(G,H)]}\right|\right|_2 \\ &= [39.702,\ 66.639] \end{aligned}$$

TABLE XII: Interval midpoints of Clusters - Grouped Cluster (((C, D), F), E) .

$\overrightarrow{[w]}_{[(A,B)]}$ =	[0.930, 0.936]	[-16, -11]	[181, 206]	[153, 196.5]
$\overrightarrow{[w]}_{[((((C,D),F),E)]}$ =	[0.916, 0.91876]	[-12, -6.625]	[85.125, 92.125]	[188.25, 194.38]
$\overrightarrow{[w]}_{[(G,H)]}$ =	[0.859, 0.867]	[26, 35]	[46.5, 62.5]	[190, 200.5]

TABLE XIII: Grouped Cluster $(((C, D), F), E)$.

	A, B	((C, D), F), E
((C, D), F), E	**[88.875, 128.11]**	
G, H	[124.14, 174.07]	**[39.702, 66.639]**

Min Distance (Linkage)

FINAL STEP

Final clustering: Starting from the new distance matrix Table XIII we have $d_{\left(\left(\left((C,D),F\right),E\right),(G,H)\right)} = [39.702,\ 66.639]$ the lowest **infimum** value of Table XIII, so we join cluster $\left(\left((C,D),F\right),E\right)$ with cluster (G,H).

Final midpoint update:

$$\overrightarrow{[w]}_{[(((((C,D),F),E),(G,H))]} = \frac{1}{2}\left(\overrightarrow{[w]}_{[((((C,D),F),E)]} + \overrightarrow{[w]}_{[(G,H)]}\right)$$

We then proceed to update midpoints Table XII into a new midpoints Table XIV, reduced in size by one row because of the clustering of $\left(\left((C,D),F\right),E\right)$ with (G,H).

Final distance matrix update:} We then proceed to update the matrix into a new distance matrix Table XV, reduced in size by one row and one column because of the clustering of (((C, D), F), E) with (G, H). Using the Table XIV, update distance between clusters is computed as:

$$d_{(((F,(C,D)),E),(G,H)\to(A,B)} = \left|\left|\overrightarrow{[w]}_{[((((F,(C,D)),E),(G,H))]} - \overrightarrow{[w]}_{[(A,B)]}\right|\right|_2 = [105.23,\ 150.13]$$

TABLE XIV: Interval midpoints of Clusters - Grouped Cluster ((((C, D), F), E), (G, H)) .

$\overrightarrow{[w]}_{[(A,B)]} =$	[0.930, 0.936]	[-16, -11]	[181, 206]	[153, 196.5]
$\overrightarrow{[w]}_{[((((C,D),F),E),(G,H))]} =$	[0.88749, 0.89288]	[7, 14.188]	[65.812, 77.313]	[189.12, 197.44]

TABLE XV: Grouped Cluster $((((C,D),F),E),(G,H))$.

	A, B
(((C, D), F), E) , (G, H)	[105.23, 150.13]

Min Distance (Linkage)

Finally, we merge the last two clusters at height $[105.23, 150.13]$. This process is summarized by the clustering diagram on Table XVI. In that diagram, the columns are associated with the objects and the rows are associated with heights of clustering. The rectangle color introduced for each level for 'X' cells is placed in a given row if the corresponding objects are merged at that stage in the clustering. We observe partial order from [heights], not every pair of height intervals are disjoint sets.

TABLE XVI: **Interval-valued Data Clustering Diagram**:IWPGMC - Midpoint update Formula

Height	A	B	C	D	E	F	G	H
[0.002,20.857]	•	•	X	X	•	•	•	•
[5, 42.06]	•	•	X	X	•	•	X	X
[11.768, 38.426]	•	•	X	X	•	X	X	X
[13, 90.632]	X	X	X	X	•	X	X	X
[14.562, 32.53]	X	X	X	X	X	X	X	X
[39.702, 66.639]	X	X	X	X	X	X	X	X
[105.23, 150.13]	X	X	X	X	X	X	X	X

V. CONCLUSION

This paper concerns the IWPGMC to be used in uncertainty quantification for interval-valued data. Note that, conclusions about the proximity of two objects can be drawn only based on the height where branches containing those two objects first are fused. Range metrics can allow for a reliable analysis of the clustering results on interval-valued data.

IWPGMC can be understood in the context of cluster membership (fuzzy clustering). Basically, it allows partial membership which means that it contains elements that have varying degrees of membership in the cluster. From this, we can understand the difference between WPGMC method and IWPGMC method: WPGMC method contains elements that satisfy precise properties of membership while IWPGMC method contains elements that satisfy imprecise properties of membership.

It is strongly recommended comparing the dendrograms from different methods and representatives (*infimum*, *midpoint*, and *supremum*) applied to several different datasets with known cluster patterns so that you can get the feel of the technique.

REFERENCES

[1] C. C. Aggarwal and C. K. Reddy, Data Clustering: Algorithms and Applications. Chapman & Hall/CRC, 1st edition, 2013.

[2] D. Shan, W. Lu, and J. Yang: Interval Granular Fuzzy Models: Concepts and Development. IEEEAccess, 24140–24153, Volume 7, 2019

[3] H. Liu, J. Li, H. Guo and C. Liu: Interval analysis-based hyperbox granular computing classification algorithms. Iranian Journal of Fuzzy Systems Vol. 14, No. 5, pp. 139–156, 2017

[4] S. M. L. Galdino, Interval-valued Data Clustering Based on the Range City Block Metric. In: SMC 2016, 2016, Budapest. The 2016 IEEE International Conference on Systems, Man, and Cybernetics, 2016.

[5] S. M. L. Galdino, and P. R. M. Maciel, Weight Pair Group Average Mean Clustering for Interval-valued Data. In: The 6th IEEE Latin American Conference on Computational Intelligence (LA-CCI). November 11-15, 2019, Guayaquil, Ecuador. The 6th IEEE Latin American Conference on Computational Intelligence (LA-CCI). Guayaquil, 2019.

[6] J. Dias and S. M. L. Galdino, Interval-valued Data Ward's Minimum Variance Clustering - Centroid update Formula. In: 2021 International Conference on Engineering and Emerging Technologies (ICEET), 2021, Istanbul. 2021 International Conference on Engineering and Emerging Technologies (ICEET), 2021. p. 1-6.

[7] R.E. Moore, Interval Analysis. Prentice Hall, Englewood Clifs, NJ, USA, 1966.

[8] U.W. Kulish and W.L. Miranker, The Arithmetic of the Digital Comput- ers: A New Approach. SIAM Review 28, 1 , 1986.

[9] R.E. Moore, R. B. Kearfott and M. J. Cloud, Introduction to INTERVAL ANALYSIS. SIAM, Philadelphia, 2009.

[10] E. R. Hansen & G. W. Walster, Global Optimization Using Internal Analysis. Second Edition, Marcel Dekker, Inc., New York, 2004.

[11] M. Ichino, General Metrics for Mixed Features - The Cartesian Space Theory for Pattern Recognition. In: Proceedings of the 1988 Conference on Systems, Man, and Cybernetics. Pergamon, Oxford, 494-–497, 1988.

[12] S.C. Johnson, Hierarchical Clustering Schemes, Psychometrika, 2:241– 254, 1967

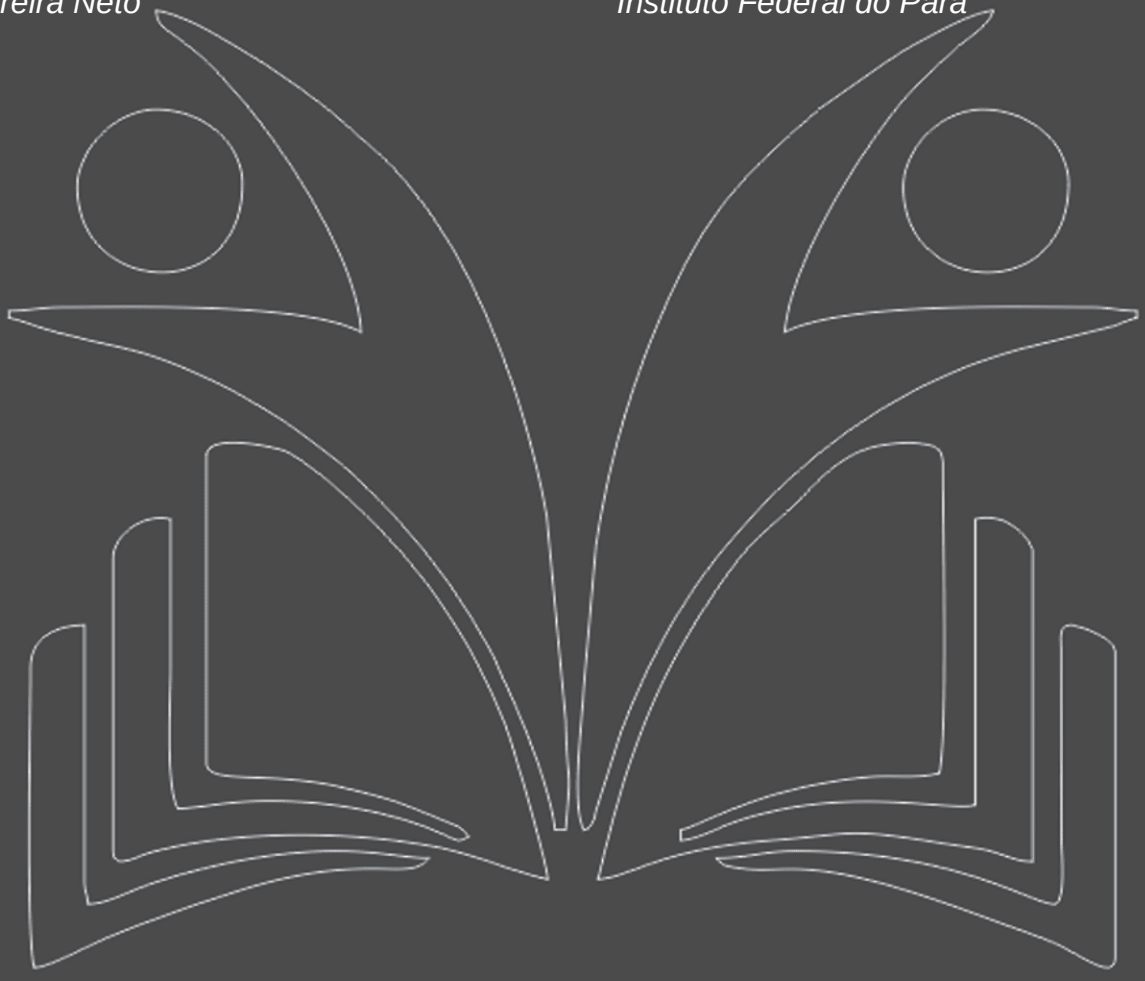
Capítulo 7

10.37423/220706252

MECANISMO DE CONTROLE DE FLUXO APLICADO NA TRANSMISSÃO DE VÍDEO NO SISTEMA LTE

IGOR MURILO BASTOS DIAS *unama*

Ananias Pereira Neto *Instituto Federal do Pará*



Resumo. *Este trabalho relata um estudo comparativo de métricas de Qualidade de Experiência (QoE) em sistema LTE (Long Term Evolution) para diferentes valores de GoP (Group of Pictures), utilizando o controle de fluxo e mapeamento de QoS (Quality of Service) entre a eNB (E-UTRAN Node B) e o UE (User Equipment), baseado na busca de solução ótima do mecanismo QFC-LTE (QoE Flow Control LTE). As comparações realizadas neste trabalho abordam as métricas: PSNR (Peak Signal to Noise Ratio), VQM (Video Quality Metric) e SSIM (Structural Similarity Index). Os resultados obtidos a partir das simulações mostram o impacto e os benefícios da proposta em comparação aos métodos convencionais.*

1. INTRODUÇÃO

Atualmente, há uma gama de serviços de telecomunicações que transmitem voz, vídeo e dados através de complexas redes de transmissão, sendo que em alguns casos, o serviço não tem uma qualidade aceitável pelo usuário final [Begazo *et al* 2011].

Nas últimas décadas, as redes móveis têm permitido grandes avanços e mudanças nos sistemas de comunicação, garantindo maior mobilidade e disseminação de conteúdo. A Tecnologia LTE (*Long Term Evolution*) é a evolução da tecnologia de rede móvel padronizada pelo 3GPP (*3rd Generation Partnership Project*), que utiliza novos sistemas de múltiplo acesso a interface aérea, para prover novos serviços e aplicações de conteúdo, tais como, MMOG (*Multimedia Online Gaming*), TV Móvel,

Web 2.0, *streaming* de vídeo, download de músicas e muitos outros [McQueen 2009].

As redes de celular de próxima geração foram criadas para facilitar o tráfego de comunicações de dados em alta velocidade, além das chamadas de voz. Novos padrões e tecnologias estão sendo implementadas para permitir que redes sem fio substituam linhas de fibra óptica ou cobre entre pontos fixos afastados por vários quilômetros (acesso sem fio fixo) [Rappaport 2009].

Um dos desafios na concepção das redes de próxima geração consiste no desenvolvimento de arquiteturas que viabilizem tanto a continuidade do serviço quanto a visão centrada no usuário, por meio do suporte adequado na QoE para aplicações multimídia, e que proporcionem sempre a melhor conectividade aos usuários móveis [Silva *et al* 2010].

A proposta deste trabalho é avaliar o desempenho da tecnologia de rede móvel LTE, baseada em métricas de QoE na disseminação de conteúdo multimídia. Avaliar a transferência de vídeos para diferentes valores de congestionamento e GoP, utilizando o controle de fluxo e mapeamento de QoS na rede de acesso LTE, compreendido entre a eNB (*E-UTRAN Node B*) e o UE (*User Equipment*). O método utiliza, dentre outros critérios, parâmetros da aplicação de Vídeo, fluxo de dados FTP e VoIP, além da relação do número de UEs e as condições de congestionamento do sistema LTE.

O artigo está organizado da seguinte forma: Na seção 2 são discutidos os trabalhos relacionados. A seção 3 apresenta uma abordagem sobre a rede LTE. As métricas de Qualidade de Experiência são mostradas na seção 4. A seção 5 apresenta o mecanismo de controle de fluxo baseado na formulação matemática do QFC-LTE. As simulações e resultados finais são descritos na seção 6, enquanto a seção 7 apresenta as conclusões e trabalhos futuros.

2. TRABALHOS RELACIONADOS

Nesta seção são apresentadas algumas propostas de mecanismos de controle de fluxo, a fim de promover uma melhor qualidade do tráfego de vídeo, levando em consideração as características deste tipo de fluxo.

Em [Begazo *et al* 2011] são realizados testes com um cenário de simulação de rede, onde são parametrizadas diferentes condições de rede, considerando um serviço de vídeo *streaming*, para o qual são consideradas diferentes qualidades de vídeo, utilizando controle de fluxo convencional. Os resultados mostram que a qualidade de vídeo é afetada diretamente pelos fatores como, o atraso e a perda de pacotes.

A proposta apresentada em [Hong *et al* 2010], desenvolveu um método que realiza a transmissão do fluxo de vídeo, denominado SAPS (*Significance Aware Packet Scheduling*), que ajustas os intervalos de tempo entre os pacotes baseado na significância do conteúdo. O trabalho compara os resultados com algoritmos utilizados na literatura, ambos com controle de fluxo *Drop Tail*. O SAPS apresentou uma melhor significância na QoE, quando comparado aos métodos tradicionais.

No trabalho desenvolvido em [Qiu *et al* 2010] é apresentado um controle de fluxo baseado na HVQ (*Hirarchy Virtual Queue*), o mecanismo propõe uma hierarquia de fila virtual evitando a perda de pacotes no sistema LTE. Além disso, o artigo investiga o mecanismo HVQ para diferentes classes de tráfego como, VoIP, Vídeo, *Best Effort*. Os resultados obtidos mostram que o mecanismo HVQ realiza o controle de fluxo com uma quantidade baixa de recursos, no entanto, produz melhor desempenho para o sistema LTE.

3. REDE LTE

A rede LTE é um padrão de tecnologia de banda larga móvel padronizada pela 3GPP, e está sendo desenvolvida para ser utilizada pelas operadoras de redes móveis como evolução das redes 3G (Terceira Geração), resultando, assim, nas Redes 4G (Quarta Geração) e foi projetada para suportar de forma eficiente vários tipos de serviços, em especial orientados a pacotes como, por exemplo, o VoIP. A arquitetura LTE está dividida, segundo [Holma e Toskala 2009], em quatro principais domínios de alto nível: UE (*User Equipment*), E-UTRAN (*Evolved UTRAN*), EPC (*Evolved Packet Core*) e *Services*.

4. MÉTRICAS DE QOE

O PSNR é uma métrica objetiva que compara a qualidade do vídeo recebido pelo usuário em relação ao vídeo original. O PSNR, desta forma, determina o valor máximo de luminosidade para cada pixel, comparando a taxa de erro do vídeo recebido.

É formulado matematicamente a partir da equação (1), para um quadro original codificado a 8 bpp (256 níveis de cinza) e expresso em dB [Rodríguez e Bressan 2012]:

$$PSNR = 10\log_{10}\left[\frac{255^2}{MSE}\right] \qquad (1)$$

Conforme cita [Rodríguez e Bressan 2012], a métrica SSIM, diferentemente do PSNR, que somente compara a taxa de erro do vídeo recebido em relação ao vídeo original, avalia o vídeo recebido considerando outros fatores, como o HVS (*Human Visual System*). Surgindo em função de o HVS ser altamente eficiente em extrair informações visuais das imagens/vídeos e não das taxas de erro, o SSIM analisa a similaridade, luminosidade, contraste e estrutura. Os valores extraídos do *frame* reconstituído pelo usuário e do *frame* original são armazenados em vetores, separadamente, sendo um vetor para cada cor. Posteriormente, obtém-se a média de cada vetor, e a combinação dessas três médias gera o valor do SSIM, indicando a qualidade do vídeo [Xinbo *et al* 2009]. O valor do SSIM é obtido pela equação (2):

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_x\sigma_y + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \qquad (2)$$

onde apresenta a função de comparação de brilho na primeira parcela da equação, μ representa a intensidade de brilho médio das imagens *x* e *y*, a função comparação contraste é representado pela segunda parcela da equação, onde representa o desvio padrão da amostra do brilho das duas imagens *x* e *y*, além da função de comparação de estrutura que relaciona a covariância de amostras de luminosidade dos quadros.

A métrica VQM, assim como as demais, baseia-se na comparação do vídeo recebido pelo usuário em relação ao vídeo original, comparando o brilho e o contraste.

Conforme afirma [Gelonch *et al* 2010].

5. MÉTODO DE CONTROLE DE FLUXO QFC-LTE

A proposta explorada neste trabalho propõe o QFC-LTE (*QoE Flow Control LTE*), capaz de realizar controle de fluxo e mapeamento de QoS entre a eNB e o EU baseado na busca de solução ótima da função objetivo $f_{QFC\text{-}LTE}(rv_n, ue_n, C_n, rb_n)$. QFC-LTE é matematicamente formulado como em (3):

$$f_{QFC-LTE} = (rv_n \times ue_n) + [(1-\lambda) \times (rb_n \times ue_n)] + C_n \qquad (3)$$

onde *rvn* representa a quantidade de recurso de Vídeo e VoIP destinado a uma conexão *n*, ue_n equivale ao número de UEs no sistema LTE destinado a *n* conexões, rb_n representa quantidade de recurso de BE (*Best Effort*) destinado a uma conexão *n* e C_n representa o nível de congestionamento na rede. O fator λ é uma variável que assume o valor 1 se a conexão é alocada somente para os tráfegos prioritários, e recebe 0, caso a conexão é alocada para todos os tráfegos.

Com base no conjunto de requisitos da função objetivo, o controle de fluxo é determinado e baseado na solução ótima esperada. Este processo é atingido quando sem consiste em maximizar a função objetivo dada por (4):

$$\max \quad f_{QFC-LTE}(rv_n, ue_n, C_n, rb_n) \qquad (4)$$

Considerando este método de controle de fluxo, quando um novo fluxo é estabelecido ocorre um novo mapeamento de QoS entre a eNB e os UEs, otimizando desta forma as aplicações de Vídeo, VoIP e FTP e os fluxos são classificados e escalonados conforme a solução ótima do método proposto.

6. SIMULAÇÕES E RESULTADOS

Para avaliar esse trabalho, foram realizadas simulações conduzidas no módulo LTE, no NS-2 (*Network Simulator* 2), versão 2.33, desenvolvido por [Qiu *et al* 2009].

Para transmissão de vídeos foi utilizado o *framework* Evalvid [Evalvid 2012]. O modelo de topologia simulado para a rede LTE está ilustrado na Figura 1.

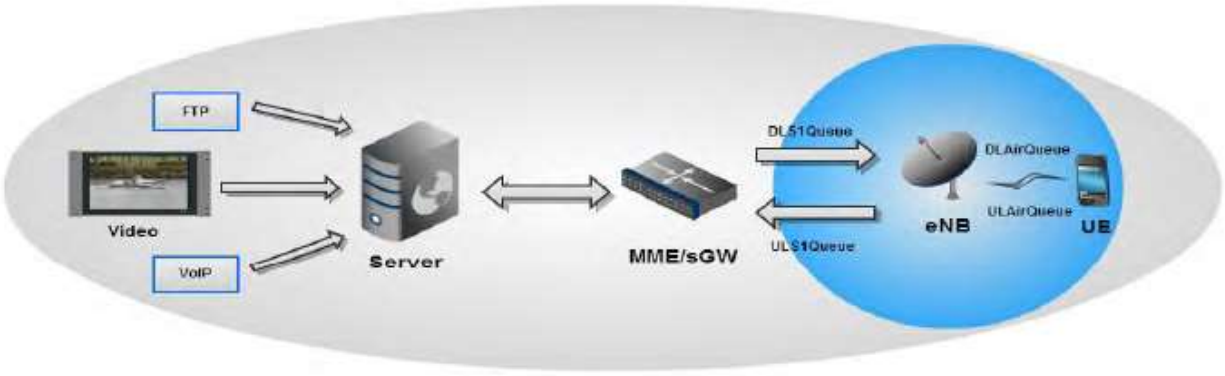


Figura 1. Cenário da Simulação para o sistema LTE.

Na simulação, o fluxo de vídeo utilizado foi o *Coastguard*, taxa de dados de 512 kbps, composto de 596 quadros, resolução de 352 x 288, 30 fps e GoP: 5 e 30. Além das especificações e padrões de vídeo, FTP e VoIP, é apresentado na Tabela 1 os principais parâmetros utilizados na simulação. Os parâmetros mostrados nesta tabela estão relacionados ao ambiente de simulação proposto em [Qiu *et al* 2010]. Além das especificações e padrões de vídeo, FTP e VoIP, a Tabela 1 apresenta os principais parâmetros utilizados na simulação. Os parâmetros mostrados nesta tabela estão relacionados ao ambiente de simulação proposto em [Qiu *et al* 2010].

Tabela 1. Parâmetros de simulação.

Parâmetros	*Valor Utilizado*
Vídeo	Taxa 512 kbps
Resolução	352 x 288
Total de Quadros	596 quadros
Taxa de Frames	30 fps
GoP	5 e 30
FTP	Taxa 512 kbps
VoIP	Taxa 8 kbps
Tempo de Simulação	19.9 s
Número de Simulações	100
Intervalo de Confiança	95%
Framework de Vídeo	Evalvid
Simulador	NS-2 2.33 e Módulo LTE

Para as métricas coletadas na simulação, utilizou-se um nível de confiança de 95% para análise dos resultados, além de analisar os dados quantitativos das métricas de QoE para PSNR, SSIM e VQM em sua média, desvio padrão e variância.

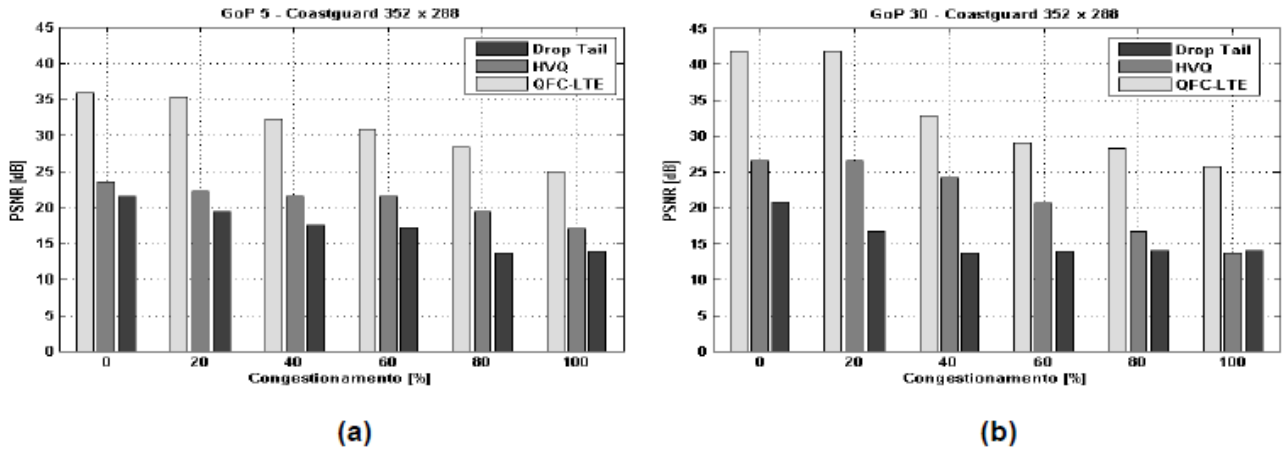


Figura 2. PSNR para fluxo de vídeo Coastguard 352 x 288: (a) GoP 5 e (b) GoP 30.

A primeira análise realizada foi quanto a métrica de PSNR, considerando diferentes níveis de congestionamento (0 – 100%) durante o processo de transmissão do vídeo *Coastguard* no sistema LTE, conforme é ilustrado na Figura 2 (a) GoP 5 e Figura 2 (b) GoP 30.

Os resultados demostram que o controle de fluxo QFC-LTE, entre a eNB e o UEm foi mais efetivo que o controle de fluxo *Drop Tail*, apresentando uma excelente qualidade de vídeo, para o QFC-LTE, segundo mapeamento de valores de PSNR para MOS (*Mean Option Score*), Tabela 2 [Piamrat *et al* 2009].

Tabela 2. Mapeamento do PSNR para o MOS.

PSNR (dB)	> 37	31 – 37	25 – 31	20 – 25	< 20
MOS	Excelente (5)	Bom (4)	Regular (3)	Pobre (2)	Péssimo (1)

Para o controle de fluxo proposto QFC-LTE, a qualidade do vídeo apresentou um nível de Bom à Regular para o GoP 5 e de Excelente à Regular para o GoP 30, devido ao mapeamento de QoS entre a eNB e o UE baseado na busca de solução ótima da função objetivo proposta.

Comparando os níveis de PSNR para o GoP 5, Figura 2 (a), o pior caso foi para controle de fluxo *Drop Tail*, a diferença máxima atingiu 52,17% em 80% de congestão, comparado ao melhor caso, QFC-LTE. Para o GoP 30, Figura 2 (b) o nível de PSNR para o QFC-LTE apresentou ganho no melhor caso, de 60,15% em 20 % de congestão comparado ao pior caso para o controle de fluxo *Drop Tail*.

Na segunda análise realizada para o controle de fluxo quanto ao nível da métrica SSIM. A Figura 3 (a) GoP 5 e Figura 3 (b) GoP 30 apresentaram as distribuições dos níveis da métrica SSIM em relação à variação do congestionamento, para os controles de fluxo QFC-LTE, HVQ e *Drop Tail*.

A Figura 4 (a) GoP 5 e Figura 4 (b) GoP 30 é ilustrado as distribuições dos níveis da métrica VQM, para as mesmas condições de congestionamento na rede LTE das métricas anteriores. Os resultados comprovam que o controle de fluxo entre a eNB e o UE, provido pelo QFC-LTE é mais eficaz que os controles HVQ e *Drop Tail*.

Comparando o melhor caso, QFC-LTE com o nível de VQM para o GoP 5, pior caso, *Drop Tail*, a diferença máxima foi de 58,10% sem congestionamento, conforme ilustra a Figura 4 (a). Nota-se que, a partir da congestão de 40%, para o GoP 5, o nível da qualidade de vídeo para a métrica VQM aumenta, comprovando os resultados das métricas anteriores para o controle de fluxo QFC-LTE.

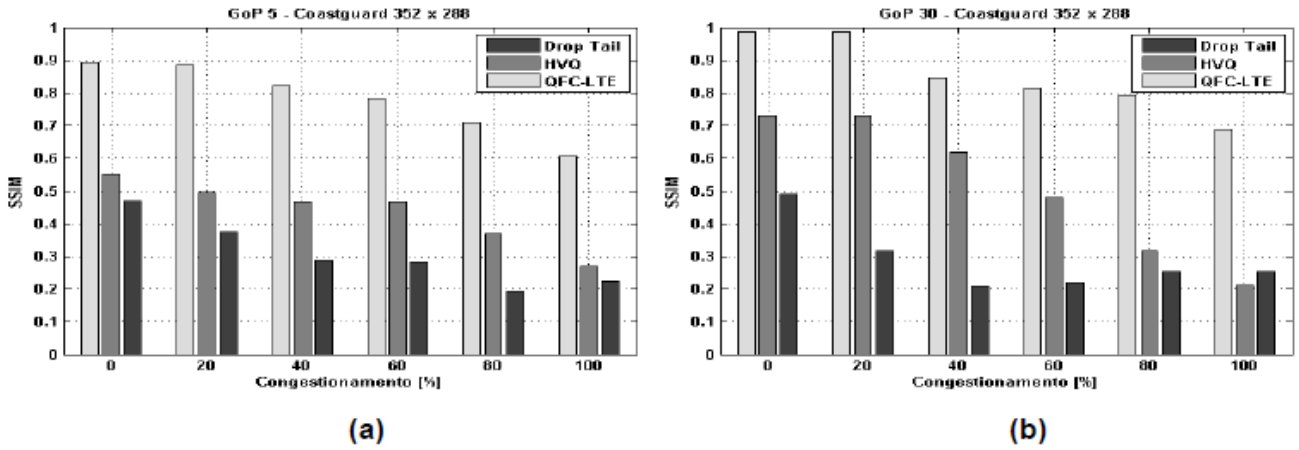


Figura 3. SSIM para fluxo de vídeo Coastguard 352 x 288: (a) GoP 5 e (b) GoP 30.

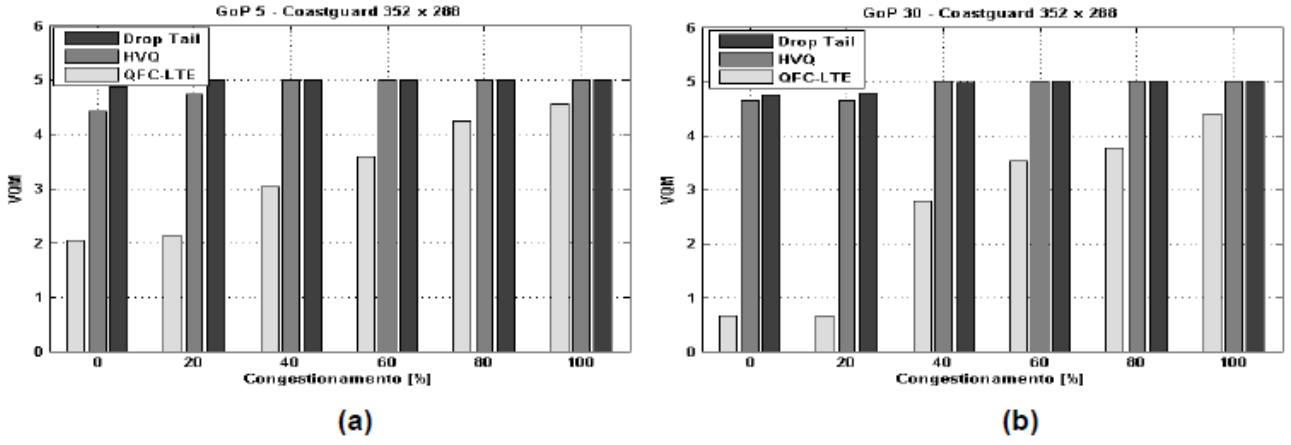


Figura 4. VQM para fluxo de vídeo Coastguard 352 x 288: (a) GoP 5 e (b) GoP 30.

Conforme a avaliação do vídeo e para validar a eficácia do controle de fluxo proposto, a Tabela 3 ilustra a sequência de quatro quadros de *frames* do vídeo *Coastguard*, *frame* 484, GoP 5 para 60% de congestionamento. Observa-se que, o vídeo recebido para o controle de fluxo QFC-LTE apresentou melhor qualidade de vídeo no fluxo entre eNB e o UE do sistema LTE.

Tabela 3. Frames do vídeo *Coastguard*, para o GoP 5 e congestão de 60%.

Frame 484 - Original	Frame 484 - QFC-LTE	Frame 484 - HVQ	Frame 484 – Drop Tail
			

Os valores médios das métricas PSNR, SSIM e VQM para os *frames* da Tabela 3 são apresentados na Tabela 4, congestionamento de 60%, o vídeo recebido de acordo com o controle de fluxo no sistema LTE, para o QFC-LTE, HVQ e *Drop Tail*, GoP 5 apresentam, respectivamente, qualificação do vídeo como Bom, Pobre e Péssimo.

Tabela 4. Resultados das métricas de QoE para GoP 5, congestão em 60%.

	PSNR [dB]	SSIM	VQM	Qualidade
QFC-LTE	31	0,78	3,58	Bom
HVQ	21,45	0,47	5	Pobre
Drop Tail	17,22	0,28	5	Péssimo

A Tabela 5 ilustra a sequência de quatro quadros de *frames* do vídeo *Coastguard*, *frame* 246, GoP 30 para 60% de congestionamento. Verifica-se que os vídeos recebidos para o controle de fluxo *Drop Tail* e HVQ foi bastante degradado. O mesmo não acontece quando o sistema LTE utilizou o controle de fluxo QFC-LTE,

Tabela 5. Frames do vídeo *Coastguard*, para o GoP 30 e congestão de 60%.

Frame 246 - Original	Frame 246 - QFC-LTE	Frame 246 - HVQ	Frame 246 – Drop Tail
			

Os valores médios das métricas PSNR, SSIM e VQM para os *frames* da Tabela 5 são apresentados na Tabela 6, congestionamento de 60% para o GoP 30, o vídeo recebido de acordo com o controle de fluxo do sistema LTE, para o QFC-LTE, HVQ e *Drop Tail*, apresentam, respectivamente, qualificação do vídeo como Regular, Pobre e Péssimo.

Tabela 6. Resultados das métricas de QoE para GoP 30, congestão em 60%.

	PSNR [dB]	SSIM	VQM	Qualidade
QFC-LTE	29	0,81	3,52	Regular
HVQ	20,64	0,48	5	Pobre
Drop Tail	13,85	0,28	5	Péssimo

Os resultados mostram que, a medida que utilizou-se o controle de fluxo QFCLTE, a perda de *frames* foi bem menor, apresentando um impacto bem melhor na qualidade do vídeo. O controle de fluxo, aliado ao mapeamento de QoS entre o eNB e o UE, melhorou as métricas de QoE, para o GoP 5 e GoP 30. Isso devido a importância do agrupamento de quadros, além do processo de atualização mais rápida dos *frames*.

7. CONCLUSÕES E TRABALHOS FUTUROS

Observamos que as métricas de QoE para a transmissão de conteúdo multimídia apresentaram melhores níveis para o controle de fluxo QFC-LTE e níveis mais baixos para os controles de fluxo HVQ e *Drop Tail*, devido a maior dependência de pacotes, codificação baixa, capacidade máxima esgotada da política de controle do *Drop Tail* (novos pacotes que chegam são descartados) e da falta de mapeamento de QoS no HVQ. A proposta QFC-LTE aumentou significantemente os níveis de QoE se comparado com os controles de fluxo HVQ e *Drop Tail* para o GoP 30, a porcentagem máxima, entre

o melhor caso, QFC-LTE e o pior caso, Drop Tail foi de 86,11% para a métrica VQM, na congestão de 20%.

Para trabalhos futuros, propõe-se um estudo utilizando outras métricas de QoE, entre elas, métricas subjetivas como, por exemplo, o MOS. Propõe-se também, a reformulação do método de controle de fluxo QFC-LTE, para considerar os diferentes valores dos GoPs para provisão de QoE no sistema LTE.

REFERÊNCIAS

Begazo, D., Rodriguez, D. e Ramírez, M. (2011) “Avaliação de Qualidade de Vídeo sobre uma Rede IP usando Métricas Objetivas”, In: Conferência Iberoamericana em Sistema, Cibernética e Informática CISCI 2011, Orlando, USA, . v. I. p. 226-229.

McQueen, D. (2009) “The momentum behind LTE adoption” IEEE Communications Magazine, vol. 47, no. 2, pp. 44–45, Feb.

Rappaport, T. S. (2009) “Comunicações Sem Fio: Princípios e Práticas”, Pearson Prentice Hall, 2ª edição, São Paulo.

Silva, D., Júnior, J., Coelho, M. e Dias, K. (2010) “Arquitetura Heterogênea com Gerenciamento da QoE e Suporte a Handover Transparente através de um Sistema Fuzzy-Genético”, In: XVI Simpósio Brasileiro de Sistemas Multimídias e Web – 2010, Belo Horizonte, MG.

Hong, S. and Won, Y. (2010) “Incorporating Packet Semantics in Scheduling of Real- Time Multimedia Streaming”, Multimedia Tool Appl., vol. 46, pp. 463-492, Jan.

Qiu, Q., Jian, C., Ping, L., and Pan, X. (2010) “Hierarchy Virtual Queue Based Flow Control in LTE/SAE”, Second International Conference on Future Networks ICFN’10, pp. 78 - 82.

Holma, H. and Toskala, A. (2009) “LTE for UMTS OFDMA and SC-FDMA Based Radio Access”, John Wiley & Sons.

Rodríguez, D. Z., and Bressan, G. (2012) “Video Quality Assessments on Digital TV and Video Streaming Services Using Objective Metrics”, IEEE Latin America Transactions, vol. 10, n. 1, Jan.

Xinbo, G., Wen, L., Dacheng, T., and Xuelong, L. (2009) “Image Quality Assessment based on Multiscale Geometric Analysis”, Image Processing, IEEE Transactions on, IEEE, v.18, n.7, pp.1409-1423.

Gelonch, A., Revés, X., Marojevic, V., Nasreddine, J., Pérez-Romero, J., and Sallent, O. (2010) “A Real Time Emulator Demonstrating Advanced Resource Management Solutions”, Wireless Personal Communications, Springer, v.54, n.1, pp. 123-136.

Qiu, Q., Jian, C., Ping, L., Zhang, Q., and Pan, X. (2009) “LTE/SAE Model and its Implementation in NS 2”, Fifth International Conference on Mobile Ad-hoc and Sensor Networks MSN’ 09, pp. 299 - 303.

Evalvid Tool (2012). Disponível em: www.tkn.tu-berlin.de/menue/research/evalvid. Piamrat, K., Viho, C., Ksentini, A., and Bonnin, J. (2009) “Quality of Experience

Measurements for Video Streaming over Wireless Networks”, Sixth International Conference on Information Technology ITNG 09, pp. 1184 - 1189.

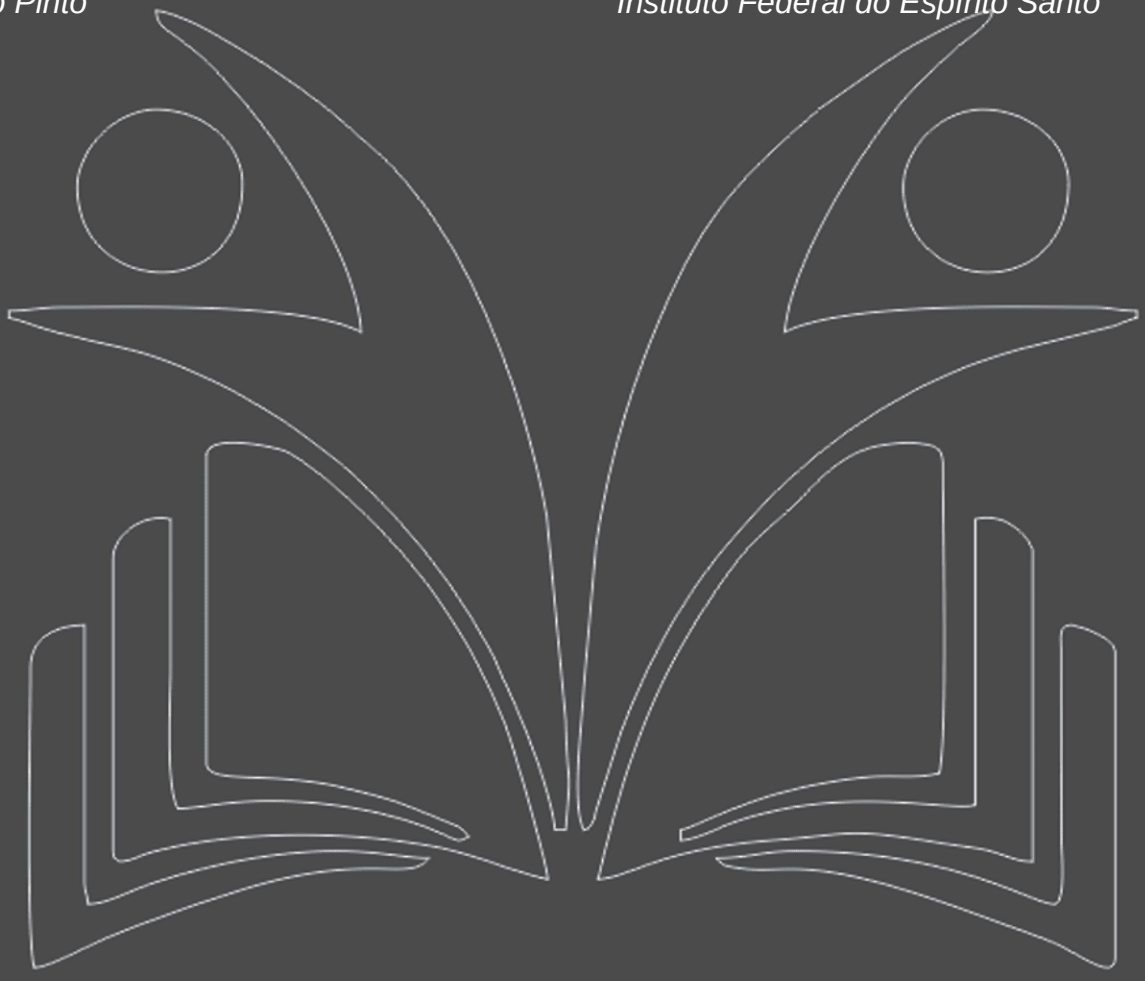
Capítulo 8

10.37423/220706275

RECONHECIMENTO AUTOMÁTICO DE FALA PARA PRODUÇÃO DE LEGENDA OCULTA UTILIZANDO CNNS E STFT

Kassius Vinícius Sipolati Bezerra — *Instituto Federal do Espírito Santo*

Luiz Alberto Pinto — *Instituto Federal do Espírito Santo*



Resumo: O reconhecimento automático de fala é um campo da computação linguística que desenvolve soluções que permitem a transcrição da linguagem falada em texto. As abordagens para esse problema evoluíram ao longo dos anos até o estado da arte de hoje, as técnicas de deep learning. Neste artigo, abordamos o assunto de reconhecimento de fala com intuito de buscar uma solução que seja melhor para os serviços de acessibilidade para a televisão, trazendo a luz uma abordagem diferente para o problema usando a classificação via redes neurais convolucionais treinadas com as imagens dos espectrogramas das STFTs dos sinais de áudio. Os resultados obtidos (acurácia de 99, 5%) indicam que a abordagem de reconhecimento de fala combinando redes neurais convolucionais e STFTs pode ser uma solução viável para o desenvolvimento de sistemas de reconhecimento automáticos de fala.

1. INTRODUÇÃO

Dados do Censo Demográfico do IBGE de 2010 mostram que 5,1% da população brasileira se auto declararam com algum grau de deficiência auditiva. Em valores absolutos, em 2010 existiam no Brasil mais de 9 milhões de deficientes auditivos. O elevado contingente de pessoas com problemas de audição, mostra a necessidade da implantação de mecanismos, legais e tecnológicos, para a promoção da acessibilidade junto a essa parcela da população, dando-lhes acesso ao consumo de conteúdo de audiovisual.

Nesse contexto, as empresas de radiodifusão, emissoras de televisão, estão obrigadas por lei a inserirem legendas ocultas em sua totalidade do conteúdo audiovisual exibido, inclusive comerciais, respeitando os critérios de assertividade e tempo de legendagem.

Legenda oculta é definida pela Agência Nacional de Telecomunicações - Anatel) como a transcrição, em língua portuguesa, dos diálogos, efeitos sonoros, sons do ambiente e demais informações que não poderiam ser percebidos ou compreendidos por pessoas com deficiência auditiva.

De acordo com a NBR 15290 [ABNT 2016], para qualquer conteúdo exibido de uma fonte gravada, a taxa de erro da transcrição deve ser de no máximo 1, 5%, enquanto para os conteúdos veiculados ao vivo, uma pequena margem de 5% de erro é aceitável.

Embora o melhor sistema de Reconhecimento Automático de Fala - Automatic Speech Recognition (ASR), disponível atualmente no mercado, fornecido pelo Google, atinja a taxa de erro estabelecida pela NBR, a forma com que a fala é processada interfere no resultado da inteligibilidade do texto. Isso porque, al´em da extração das informações fonéticas, o ASR desenvolvido pelo Google também utiliza a análise do modelo de linguagem, que considera os aspectos semânticos e gramaticais, bem como as informações contextuais para a construção das sentenças [Aleksic et al. 2015]. A aplicação desse modelo de linguagem permite que palavras em qualquer posição da frase sejam alteradas para se ajustar ao contexto do discurso, o que resulta em um reconhecimento de fala mais próximo do natural.

Contudo, a troca de palavras reduz o tempo de exposição do texto para leitura.

Para ilustrar, um telejornal local no estado do Espírito Santo deu a seguinte notícia: Chikungunya está assustando os moradores de Cachoeiro, no momento inicial o ASR legendou como, Chico Cunha está assustando os moradores de Cachoeiro. Somente após a análise do contexto a notícia foi legendada corretamente. Contudo, o tempo de exposição da primeira frase foi de aproximadamente 3,5 segundos. Após a troca da palavra pelo ASR o tempo de leitura para a frase correta foi de

aproximadamente 0,5 segundo. Se considerarmos uma aplicação de reconhecimento de fala para inserção de legendas, tendo em vista a acessibilidade de pessoas com deficiência auditiva, a redução do tempo de leitura pode impedir sua completa acessibilidade ao evento.

Os critérios da NBR para validação de ASR para legendagem utiliza duas métricas, a WER (1) e o NER (2). Enquanto a WER (Word Error Rate) é uma forma de avaliação da quantidade de erros na legenda, o NER (Number of words, Edition Error e Recognition error) avalia a qualidade do que está sendo transcrito.

$$WER = \frac{(N - E)}{N} \qquad (1)$$

Onde:

N: Quantidade de palavras ditas pelo interlocutor;

E: Quantidade de palavras transcritas incorretamente.

$$NER = \frac{(N - E - R)}{N} \qquad (2)$$

Onde:

N: Quantidade de palavras ditas pelo interlocutor, pontuações, identificações de interlocutores e pausas;

E: Erros de edição, geralmente causados pelas más escolhas do legendista/sistema na produção do CC (Closed Caption);

R: Erros de reconhecimento causados por pronúncia ou escuta incorreta. Também podem ser causados pela tecnologia usada para produzir as legendas. Esses erros podem ser inserções, exclusões ou substituições.

Quando considerado a WER, os melhores ASRs disponíveis no mercado atendem os critérios da NBR. Contudo, os valores do NER desses sistemas ficam abaixo dos estabelecidos pela norma.

Nesse contexto, esse trabalho propõem uma estrutura para ASR que faça a transcrição da fala do interlocutor, em tempo real, enfatizando a estrutura fonética do texto, tendo em vista eliminar os erros de reconhecimento e aumentar o tempo de exposição para leitura do texto legendado por pessoas portadoras de deficiência auditiva.

A avaliação de desempenho do sistema proposto utilizou como métrica o NER, tendo em vista que essa métrica já considera o número de palavras classificadas incorretamente, bem como os erros de edição, como as trocas de palavras ao longo da criação da legenda pelos sistemas de ASR atuais.

Para a implementação do sistema, os espectrogramas obtidos das Short-Time Fourier Transform - STFT dos sinais de áudio de um conjunto de dados de teste foram convertidos em imagens que, posteriormente foram fornecidas como entradas para uma Rede

Neural Convolucional (Convolutional Neural Networks - CNN) [Krizhevsky et al. 2017], topologia de rede reconhecida pelos excelente desempenho em tarefas de classificação de imagens. O desempenho do sistema STFT-CNN foi comparado a um método de referência, que combina os descritores dos sinais de fala extraídos pelas Mel-Frequency Cepstral Coefficients - MFCC e uma CNN. A MFCC é uma técnica consolidada, visto que suas etapas são modeladas para aproximar o sistema da forma com que o ouvido humano funciona.

2. O SINAL DE VOZ

Devido as constantes alterações no formato do trato vocal, a forma da onda do sinal de voz também varia com o tempo, sendo considerado, para efeito de processamento, não-estacionário. Sinais não-estacionários são caracterizados por apresentarem rápidas alterações das propriedades acústicas e espectrais em pequenos intervalos de tempo. O processamento de sinais dessa natureza pode fazer uma consideração de estacionariedade, nesse caso, o sinal será subdividido em segmentos de curta duração que tenham características aproximadamente estacionárias. Uma outra alternativa seria a utilização de técnicas de processamento de sinais não-estacionários [T¨uske et al. 2012].

Na linguística, os fonemas são a unidade elementar da fala e as suas características acústicas e espectrais diferem de língua para língua. Um fone é o menor segmento discreto perceptível de som em uma corrente da fala [Crystal 1988]. A realização desse segmento difere de acordo com diversos fatores, como por exemplo sexo, idade e região. Assim, associado a cada fonema está um conjunto de fones com ligeiras variações acústicas, que se denominam alofones. A maioria das palavras é composta por mais de um fonema, e os fonemas diferem entre si na duração, no tipo de excitação e no posicionamento dos diferentes articuladores durante a sua produção. A transição na mesma palavra de um fonema para um outro é feita de forma contínua, o que implica que a transição das propriedades acústicas e espectrais entre fonemas também varia continuamente.

As fases de transição entre fonemas acontecem porque ao transitar de um fonema para outro é necessário reposicionar os articuladores de modo a alterar o formato do trato vocal para produzir o novo fonema. Como esse reposicionamento é realizado pelos músculos, é impossível fazê-lo de forma instantânea. Por este motivo, o mesmo fonema inserido em duas palavras distintas pode não ter exatamente nem a mesma duração nem as mesmas características sonoras e espectrais, pois tais características dependem do fonema anterior e do fonema posterior.

Nos sinais de voz, quando as palavras são pronunciadas rapidamente, e os fones são de curta duração, a posição final dos articuladores ao produzir um determinado fonema pode não ser atingida, pois a fase de transição desse fonema com os fonemas, anterior e posterior estão parcialmente sobrepostas.

2.1. MODELAGEM DO SINAL DE FALA

Como os sinais de voz são não-estacionários, seus parâmetros estatísticos e sua forma de onda se alteram ao longo do tempo. Estas alterações se devem `as modificações dos articuladores envolvidos no processo fonador. As ferramentas utilizadas no processamento desses sinais requerem que os mesmos permaneça m invariantes no tempo. Na produção da voz estão envolvidos diferentes órgãos, ossos e músculos e, devido à inércia destes articuladores, não é possível alterar as suas posições de forma abrupta nem instantaneamente.

Modificar o posicionamento dos diversos articuladores e consequentemente alterar a forma do trato vocal é, portanto, um processo contínuo e suave. Dessa forma, se um sinal de voz for dividido em segmentos de duração suficientemente curta (aproximadamente 20ms), estes segmentos de curta duração poderão ser considerados quase estacionários, pois durante a sua duração os articuladores movem-se pouco e lentamente para que as características acústicas do segmento possam ser consideradas invariantes no tempo [Goldberg and Riek 2000] .

Vários métodos foram propostos para a extração de descritores de sinais de voz, como, por exemplo, o Linear Predictive Coding - LPC [Sanjaya et al. 2018], o Perceptual Linear Prediction - PLP [Hermansky 1990] e o Mel-Frequency Cepstral Coefficients - MFCC. Uma abordagem de uso mais geral em processamento de sinais, considera a utilização da STFT para a extração de descritores dos sinais de fala. A propriedade da STFT de realizar a análise dos sinais, simultaneamente, nos domínios do tempo e da frequência pode apresentar vantagens sobre as técnicas anteriores.

Os três primeiros métodos citados anteriormente são os de uso mais comum em tarefas de reconhecimento automático de fala. Devido a sua natureza linear, onde se assume um mesmo peso

para todo o espectro da fala, o LPC torna-se uma opção menos atraente. O PLP e o MFCC são baseados no conceito dos filter banks logaritmicamente espaçados, que se aproximam da forma da escuta humana, mostrando-se boas opções para a tarefa de ASR [Dave 2013]. Contudo, devido a sua capacidade para extração de informações nos domínios do tempo e da frequência, simultaneamente, nesse trabalho o método para extração de descritores a ser utilizado será a STFT.

3. SHORT-TIME FOURIER TRANSFORM - STFT

A Short-Time Fourier Transform é uma função linear do sinal x[m] que, de acordo com a Equação 3 pode ser descrita como a transformada de Fourier de um segmento de janela do sinal, x[m] ∗ w[n − m]. Se a janela de analise w[n], for real e par, a STFT será a tranformada de Fourier de um segmento do sinal centralizado no momento da saída m, [Papandreou-Suppappola 2018].

$$X(n,\omega) = \sum_{m=-\infty}^{\infty} x[m] * w[n-m] * e^{-j\omega n}. \quad (3)$$

A Equação 4 demonstra que a STFT também pode ser vista como uma versão do sinal após a aplicação de um filtro. Se a janela de analise for uma função do tipo passa-baixa, o sinal ´e enviado através de um filtro passa-banda.

$$X(n,\omega_0) = \sum_{m=-\infty}^{\infty} (x[m] * e^{-j\omega_0 m}) * w[n-m] \quad (4)$$

Um dos aspectos importantes da utilização da STFT, é a sua representação gráfica na forma de espectrograma. Para as representações em tempo e frequência o espectrograma será relacionado com o módulo quadrático da STFT, como mostra a Equação 3, [Das 2015].

$$SPEC_x = |X(n,\omega)|^2 = \left| \sum_{m=-\infty}^{\infty} x[m] * w[n-m] * e^{-j\omega n} \right|^2 =$$

$$= \left| \sum_{m=-\infty}^{\infty} (x[m] * e^{-j\omega_0 m}) * w[n-m] \right|^2 \quad (5)$$

Pode-se destacar como vantagem da aplicação da STFT a facilidade de interpretação dos resultados e a baixa complexidade computacional para sua implementação, além disso a STFT tem sido amplamente utilizado na análise de sinais quasi-estacionários.

Entre as desvantagens de sua aplicação podem ser mencionadas, seu comprimento no tempo e sua largura de banda maiores que a do sinal original, dessa forma, faz-se necessário a complementação de sinal na representação tempo-frequência (padding). Além disso, existe uma relação de compromisso na representação nos domínios do tempo e da frequência, não sendo possível obter boa resolução em ambos os domínios simultaneamente, [Poularikas 2010].

4. REDES NEURAIS CONVOLUCIONAIS

As Redes Neurais Convolucionais (Convolutional Neural Network - CNN) têm sido utilizadas com sucesso em aplicações de classificação, detecção e reconhecimento de objetos em imagens [Girshick et al. 2013] e em vídeos. Segundo [Liu et al. 2017], além das camadas de entrada e de saída, uma rede neural convolucional é constituída por várias outras: camada convolucional, camada de pooling, camada totalmente conectada, camada de normalização em lote. Cada uma dessas camadas executam diferentes operações sobre os dados de entrada e, para o bom desempenho da rede, requerem a configuração de diferentes parâmetros. A seguir as camadas mais importantes das CNNs são descritas, considerando sua aplicação no processamento de imagens.

Camada Convolucional: essa camada é responsável pela extração das características do sinal de entrada (imagem ou sinal unidimensional), para o caso de imagens essas características poderiam ser as bordas e as cores. Na configuração dos parâmetros dessa camada deve-se definir as dimensões e a quantidade de filtros que serão aplicadas, o stride da convolação, que é a quantidades de saltos do filtro ao se deslocar pela imagem e o padding que define como será o tratamento das bordas da imagem na realização da convolação. O número de filtros corresponde ao número de neurônios que conectam àmesma região da entrada, e determina as dimensões do mapa de características. As redes convolucionais, em geral, possuem várias camadas convolucionais que são capazes de extrair informações em níveis, sucessivamente, mais baixos de detalhamento.

Função de Ativação: em redes neurais a função de ativação, de acordo com o resultado da soma ponderada das entradas de um nó precedente, efetua a ativação de um nó consequente de saída. ´ arias funções de ativação podem ser utilizadas tais como sigmoid, tanh, softmax e ReLU. Entre essas, a que tem sido utilizado como mais sucesso em redes neurais convolucionais é a ReLU (Rectified Linear Unit). A ReLU é uma função de ativação linear por partes, cuja saída será igual a entrada se a mesma for positiva, ou, caso contrário, a saída será igual a zero, de acordo com a Equação 6. Pela facilidade

de treinamento e por, de forma geral, resultar em bom desempenho das redes, essa função de ativação tem sido muito utilizada em tarefas que utilizam as CNNs.

$$f(x) = \begin{cases} x, x \geq 0 \\ 0, x < 0 \end{cases} \quad (6)$$

Camada de Pooling: a camada de pooling, geralmente, está localizada após uma camada de convolucão. Sua principal função é a redução da dimensão espacial da imagem de entrada (comprimento e altura) para a camada convolucional seguinte. Na prática, essa camada realiza uma subamostragem dividindo a entrada em regiões retangulares menores e calculando o valor correspondente de cada região. Esta operação reduz o tamanho do mapa de características removendo informações de redundância espacial. Uma das principais vantagens da utilização da camada de pooling é a possibilidade de aumentar o número de filtros de convolução nas camadas mais profundas sem que isso resulte no aumento considerável do esforço computacional. A abordagem mais comumente utilizada para a implementação da camada de pooling é o Max Pooling. Nesse caso, no processo de subamostragem será considerado o maior valor da região. De forma análoga, uma camada Average Pooling realiza a subamostragem calculando o valor médio da região ao invés do valor máximo, sendo possível ainda a implementação do Min Pooling, nesse caso o menor valor da região será selecionado.

Camada de Normalização em Lote: a Batch Normalization Layer é utilizada para reduzir a faixa de variação dos valores das unidades das camadas ocultas, o que resulta na aceleração do treinamento da rede e na redução da sua sensibilidade aos pesos iniciais, além de possibilitar que cada camada da rede aprenda de forma independente das demais [Ioffe and Szegedy 2015].

Camada Totalmente Conectada: a camada totalmente conectada (Fully Connected Layer) é localizada antes da camada de saída da rede e possui todos os seus neurônios conectados a todos neurônios da camada anterior, combinando, dessa forma, todas as características aprendidas pelas camadas anteriores para o mapa de características que melhor represente a imagem. Em problemas de classificação, a última camada totalmente conectada combina as características para classificar as imagens. Dessa forma, o tamanho da saída da última camada totalmente conectada ´e igual ao número de classes do dataset.

Camada de Classificação: em problemas de classificação, a utilização da função de ativação Softmax associada a camada de classificação deve vir após a última camada totalmente conectada. A função de ativação Softmax normaliza a saída da camada anterior.

Sua saída consiste de números positivos cuja soma é igual a 1. Estes números podem ser usados como probabilidades pela camada de classificação. A Camada de Classificação é a camada final da rede e infere a quantidade de classes a partir do tamanho da saída da camada anterior. As probabilidades retornadas pela função de ativação Softmax para cada entrada são utilizadas para uma classe entre as várias possíveis.

Considerando as diferentes configurações que podem ser obtidas para as redes CNNs pela combinação dos diferentes tipos de camadas e pela quantidade de cada uma delas, o principal esforço para a utilização dessas redes deve ser direcionado à busca pela melhor arquitetura para o problema em questão. Algumas arquiteturas de redes neurais convolucionais que têm sido propostas no decorrer dos últimos anos, têm se notabilizado pelo excelente desempenho em tarefas envolvendo imagens. Por exemplo, a LeNet-5 (LeCun et al in 1998); AlexNet [Krizhevsky et al. 2012a]; ZFNet [Zeiler and Fergus 2013], VGGNet [Simonyan and Andrew 2015], GoogLeNet [Szegedy et al. 2014], ResNet [Kaiming He 2015], ResNext [Xie et al. 2017] e SENet [Hu et al. 2018].

4.1. A ARQUITETURA DA GOOGLENET

Por causa do seu excelente desempenho em tarefas de extração de características e de classificação envolvendo imagens, nesse trabalho decidiu-se pela utilização da rede GoogLeNet.

Proposta por [Szegedy et al. 2014], a rede GoogLeNet foi a vencedora do Image-Net Large Scale Visual Recognition Challenge (ILSVRC) do ano de 2014. A GoogLeNet tem como principal característica a introdução do módulo Inception, cuja utilização reduz o número de parâmetros da rede. Quando comparado com a AlexNet, que possui em torno de 60 milhões de parâmetros, a GoogLeNet possui cerca de 4 milhões.

Uma das dificuldades para a classificação de objetos em imagens é o que se denomina problema da localização da informação. O problema da localização é ocasionado pela diferença nas dimensões que o objeto de interesse pode apresentar entre as imagens de um conjunto. Dessa forma, a escolha de um filtro de dimensões adequadas á todas as imagens não é uma tarefa simples. Filtros com maiores dimensões são adequados quando a informação está distribuída em toda a extensão da imagem. Por

sua vez, filtros com menores dimensões têm melhor desempenho quando a informação se encontra mais concentrada em uma região específica.

De acordo com [Busson et al. 2018], a introdução do módulo Inception pode minimizar os efeitos do problema da localização da informação através da utilização de filtros de diferentes dimensões na mesma camada, deixando a rede mais larga em comparação `as arquiteturas de CNNs convencionais. O módulo Inception funciona como um extrator de características, realizando convoluções com três diferentes tamanhos de filtros, 1x1, 3x3 e 5x5. A fim de reduzir o custo computacional nessa etapa, uma camada Max Pooling com um filtro 3x3 ´e adicionada e, além disso, o número de camadas da entrada é limitado utilizando convoluções 1x1 antes das convoluções 3x3 e 5x5. Apenas na camada de Max Pooling, a convolução 1x1 é realizada posteriormente.

A GoogLeNet utiliza 9 módulos Inception em sequência, possuindo um total de 22 camadas. Como ocorre em redes neurais com uma quantidade muito grande de camadas treinadas utilizando métodos baseados em gradiente, a GoogLeNet está sujeita ao problema que se denomina gradient vanishing. Como consequência, os pesos da rede são atualizados proporcionalmente `a derivada parcial da função de erro em relação ao peso atual. Nesse caso, se o gradiente for muito pequeno, não haverá alteração dos pesos.

No pior caso o treinamento da rede pode ser interrompido. Para a solução do problema do gradient vanishing foram inseridas duas sa´ıdas auxiliares em dois dos nove módulos Inception.

5. METODOLOGIA

Nessa seção são descritas todas as etapas da construção do trabalho, a base de dados utilizada para a realização dos experimentos, bem como as etapas de pré-processamento a que os dados foram submetidos. Além disso, é apresentada também a configuração do classificador e as métricas aplicadas na avaliação dos resultados.

5.1. BASE DE DADOS

Na realização do trabalho foi utilizada uma base de dados com diferentes palavras para comandos de voz. Essa base foi obtida através do site Kaggle, uma comunidade para cientista de dados e machine learning mantida pelo Google LLC. A base, denominada Synthetic Speech Commands Dataset, contém 41.849 arquivos de áudio em formato wave [Buchner 2017], consistindo de 30 diferentes palavras pronunciadas.

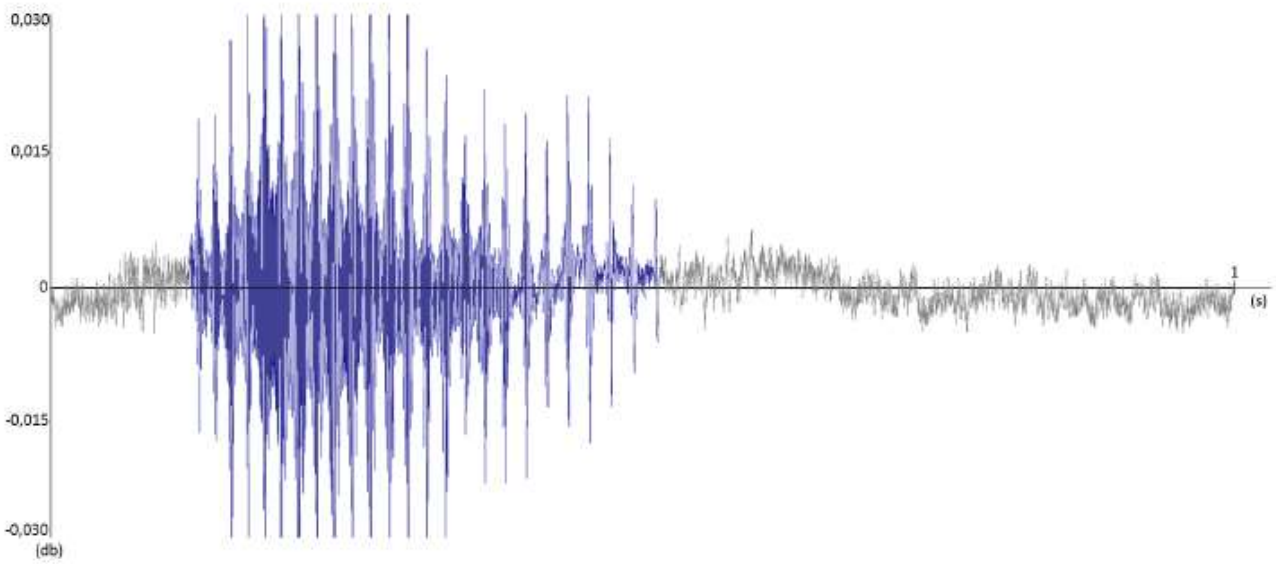
Os arquivos da base estão organizados em uma estrutura de pastas, sendo cada pasta rotulada pela palavra pronunciada. Cada pasta é constituída pelos arquivos de áudio em formato wave com a pronúncia das palavras. As palavras estão no idioma inglêss e foram produzidas sinteticamente usando a pronuncia descrita no British English Example Pronciation Dictionary [Robinson 1997]. Em cada classe, para cada arquivo de áudio, foram adicionadas variações de níveis de stress, pitch, velocidade e interlocutores.

Todos os 41.849 arquivos de áudio da base são mono (possuem apenas um canal de gravação e reprodução), com duração de 1 segundo. Possuem resolução de 16 bits e taxa de amostragem de 16.000 Hz. Apesar de terem a duração fixa, a faixa ocupada pelo som das palavras pode ser menor do que o tamanho total do arquivo. Com isso, é possível que existam trechos de silêncio antes e/ou depois de cada pronúncia. A Figura 1 mostra o sinal resultante da pronúncia da palavra five. A parte em azul indica o trecho onde a palavra foi efetivamente pronunciada e a parte em cinza mostra os silêncio antes e depois da pronúncia. Como pode ser notado, esse silêncio não representa falta completa de áudio e sim um ruído de baixa intensidade que não faz parte da informação da palavra.

Para formulação do problema de reconhecimento de padrões, cada palavra da base de dados será associada a uma classe. Dessa forma, o problema será constituído por 30 classes. Cada classe possui um número específico de amostras. A classe com menor número de amostras possui 902 e a com maior contém 2.400 amostras. A Figura 2 mostra o desbalanceamento do conjunto de dados e a quantidade de áudios para cada classe.

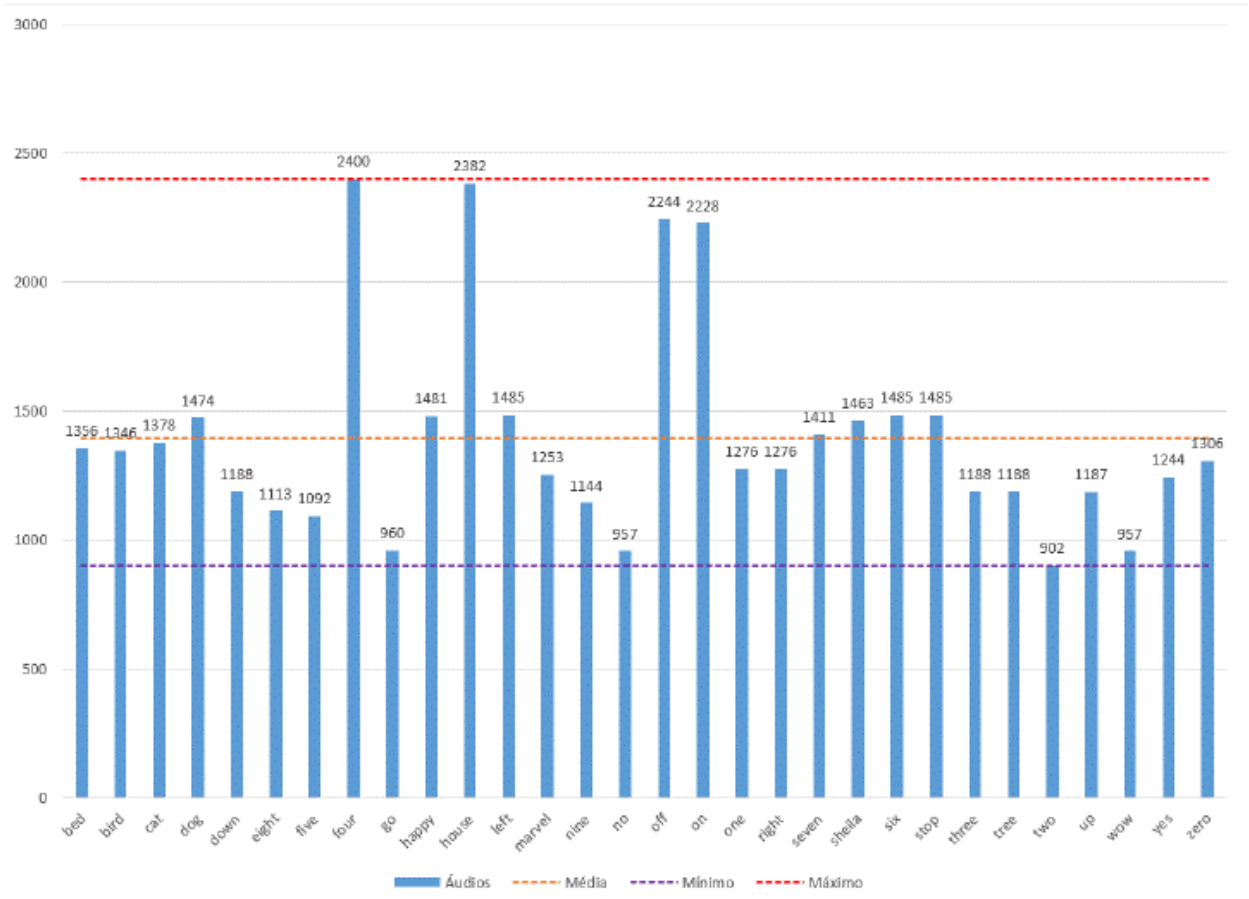
Para o treinamento e teste do classificador, a base foi dividida em 70% das amostras para treinamento e 30% para teste. Preservando a característica do conjunto original, o desbalanceamento se manteve também na divisão entre os conjuntos de treino e teste, o que pode ser observado na Tabela 1.

Figura 1. Waveform mostrando a pronúncia da palavra five



Elaborado pelo autor (2019)

Figura 2. Desbalanceamento entre as classes do dataset



Elaborado pelo autor (2019)

Além dos arquivos de áudio descritos, a base de dados contém também arquivos auxiliares semelhantes aos originais, nos quais foram adicionados ruídos de forma artificial.

A adição de ruído na etapa de treinamento pode melhorar a robustez do classificador para diferentes condições de entrada. Entretanto, para a produção de legendas ocultas as condições são sempre de baixo ou nenhum ruído. Dessa forma, o treinamento com a base de áudio limpa torna-se mais eficaz, [Hirsch and Pearce 2000].

Tabela 1. Partição das classes do conjunto de dados em treino e teste

CLASSE	TREINO	TESTE	CLASSE	TREINO	TESTE
bed	949	407	*off*	1571	673
bird	942	404	*on*	1560	668
cat	965	413	*one*	893	383
dog	1032	442	*right*	893	383
down	832	356	*seven*	988	423
eight	779	334	*sheila*	1024	439
five	764	328	*six*	1040	445
four	1680	720	*stop*	1040	445
go	672	288	*three*	832	356
happy	1037	444	*tree*	832	356
house	1667	715	*two*	631	271
left	1040	445	*up*	831	356
marvel	877	376	*wow*	670	287
nine	801	343	*yes*	871	373
no	670	287	*zero*	914	392

5.2. REMOÇÃO DE TRECHOS DE SILÊNCIO

Previamente a etapa de extração da STFT, os 41.849 arquivos de áudio foram submetidos ao processo de remoção dos silêncio. Originalmente, todos os arquivos tinham duração igual a 1 segundo. Após o tratamento, os arquivos tiveram suas durações alteradas. O volume de arquivos, a duração média e a duração total, que descreve o somatório da duração de todos os áudios de cada classe, estão representados na Tabela 2. É possível observar que algumas palavras possuem tempo de pronúncia menor que outras. Por exemplo, o tempo necessário para a pronúncia da palavra up foi, em média,

174ms. Por sua vez, para palavras com fonemas maiores como sheila, o tempo médio de fala foi 568ms.

5.3. EXTRAÇÃO DAS STFTS

A STFT é uma sequência de transformadas de Fourier de um sinal subdividido em frames, no tempo. Tarefa importante para a obtenção da STFT de um sinal é a definição do resolução no tempo. Janelas grandes melhoram a resolução na frequência, mas pioram a resolução no tempo, [Yuegang et al. 2007]. A Figura 3 mostra a diferença do espectrograma para a pronúncia da palavra five com o tamanho da janela (L) definido em 16 e 3072. Wisdom [Wisdom et al. 2015] mostrou em seu trabalho que, para a tarefa de reconhecimento de fala, a utilização de janelas maiores resulta em melhor desempenho do classificador. Para a realização desse trabalho foi adotado janelas de 1024, a mesma largura de janela utilizado em [Wisdom et al. 2015].

Além do tamanho da janela, a relação de deslizamento desta ao longo do sinal

e a função de janelamento são outros parâmetros importantes para a obtenção do espectrograma.

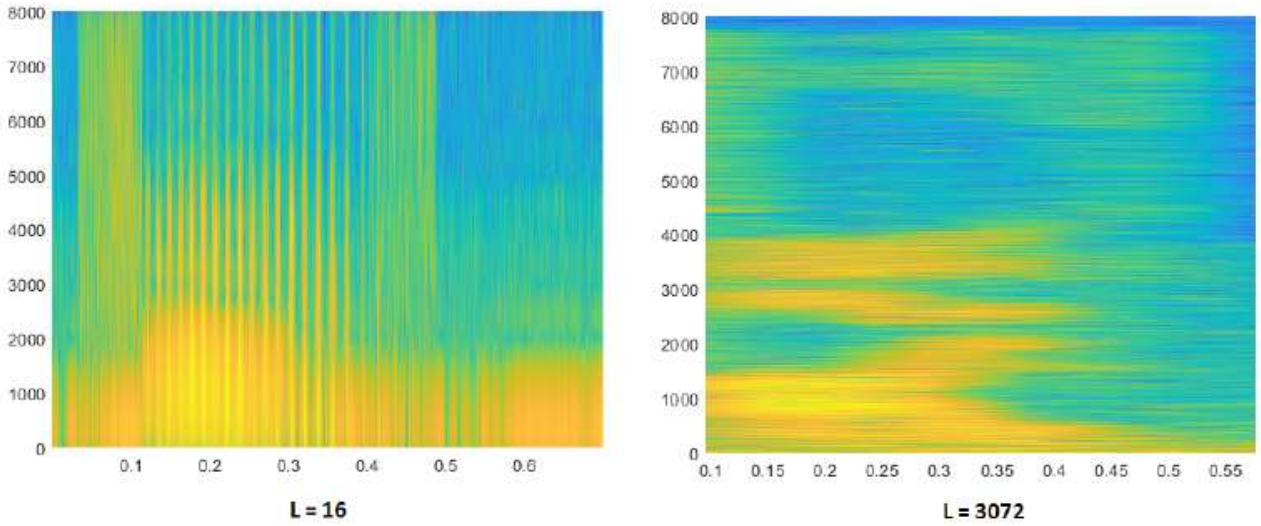
Os valores escolhidos para o deslizamento da janela foi de L/8, onde L é o tamanho da janela. Por ser amplamente utilizada, nesse trabalho foi decidido pela janela de Hanning [Paliwal and Alsteris 2005]. A Figura 4 mostra o espectrograma obtido com os parâmetros determinados. Após calculadas as STFTs de todos os arquivos de áudios do

Tabela 2. Duração da pronúncia média de cada classe do dataset após a etapa de pré-processamento.

Classes	Áudios	Duração Média (ms)
bed	1356	339
bird	1346	249
cat	1378	378
dog	1474	199
down	1188	423
eight	1113	363
five	1092	311
four	2400	273
go	960	357
happy	1481	451
house	2382	471
left	1485	484
marvel	1253	458
nine	1144	528
no	957	368

Classes	Áudios	Duração Média (ms)
off	2244	436
on	2228	212
one	1276	431
right	1276	440
seven	1411	516
sheila	1463	568
six	1485	418
stop	1485	306
three	1188	288
tree	1188	328
two	902	434
up	1187	174
wow	957	374
yes	1244	404
zero	1306	429

Figura 3. Diferença no espectrograma com diferentes tamanhos de janelas



Elaborado pelo autor

conjunto, o espectrograma de cada um dos arquivos foi obtido, e as imagens foram armazenadas, criando um novo conjunto que, foi utilizado para o treinamento da GoogLenet.

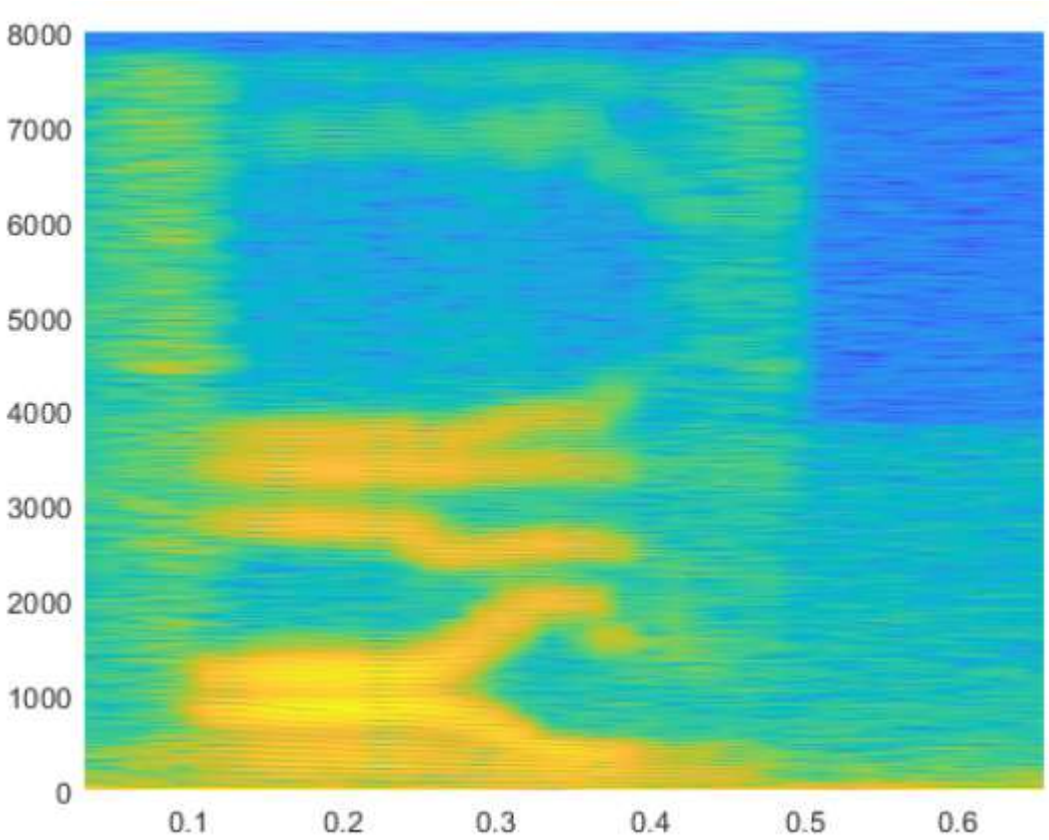
5.4. CLASSIFICAÇÃO

Para a classificação das palavras pronunciadas, a GoogLeNet foi treinada com as imagens os espectrogramas das STFTs. Todas as imagens possuem resolução igual a 224 × 224 × 3.

A rede foi treinada com oito épocas de treinamento, após o que a acurácia, a precisão, o recall e o F1-Score foram calculados [Sokolova and Lapalme 2009].

Mesmo tendo sido superada por versões posteriores, a arquitetura de rede GoogLeNet trouxe novas técnicas, diferentes das CNNs tradicionais, que foram precursoras para a evolução das redes que a superaram posteriormente. As principais diferenças na arquitetura da GoogLeNet são: ao longo das camadas existem convolucões 1 × 1; ao

Figura 4. Espectrograma da palavra five



Elaborado pelo autor

fim da rede é utilizado o global average pooling ao invés da tradicional Fully Connected

layer [Lin et al. 2013]; e os inception modules, que são empilhamento de convolucões de tamanhos e tipos diferentes para a mesma entrada.

A vantagem da utilização das convoluções 1 × 1 é a redução das dimensões do problema, o que reduz o custo computacional e permite o crescimento da rede em profundidade e/ou largura. Se considerarmos como exemplo um trecho qualquer da arquitetura da GoogLeNet para base de dados do problema, calculando uma convoluções 5 × 5(32) diretamente sem o uso da convolução 1 × 1, o número de operações necessárias seria de:

$$\mathrm{NO}_{5x5} = (5^2 \times 192 \times 32 \times 224^2) = 7,07 \times 10^9.$$

Como na construção da GoogLeNet são utilizadas as convoluções 1 × 1(16), o número de operações para esse trecho é :

$$\mathrm{NO}_{1x1} = (1^2 \times 192 \times 16 \times 224^2) = 1,541 \times 10^6,$$
$$\mathrm{NO}_{5X5} = (5^2 \times 16 \times 32 \times 224^2) = 6,422 \times 10^6,$$
$$\mathrm{NO}_{total} = NO_{1x1} + NO_{5X5} = 7,963 \times 10^6.$$

Como consequência da utilização das convoluções 1 x 1 ocorre uma redução de 6, 91 × 109 operações, apenas nessa parte da rede. A Figura 5 detalha os parâmetros de cada camada, onde as colunas identificadas como reduce são responsáveis pela redução das dimensões do problema, pois apresentam uma convolução 1×1 antes das convoluções 3 × 3 e 5 × 5.

Ao todo, o classificador possui 22 camadas se forem contadas apenas as que contêm parâmetros, e 27 se considerados as camadas de pooling. As camadas iniciais

Figura 5. Detalhamento dos Inception Modules da GoogLeNet

type	patch size/ stride	output size	depth	#1×1	#3×3 reduce	#3×3	#5×5 reduce	#5×5	pool proj	params	ops
convolution	7×7/2	112×112×64	1							2.7K	34M
max pool	3×3/2	56×56×64	0								
convolution	3×3/1	56×56×192	2		64	192				112K	360M
max pool	3×3/2	28×28×192	0								
inception (3a)		28×28×256	2	64	96	128	16	32	32	159K	128M
inception (3b)		28×28×480	2	128	128	192	32	96	64	380K	304M
max pool	3×3/2	14×14×480	0								
inception (4a)		14×14×512	2	192	96	208	16	48	64	364K	73M
inception (4b)		14×14×512	2	160	112	224	24	64	64	437K	88M
inception (4c)		14×14×512	2	128	128	256	24	64	64	463K	100M
inception (4d)		14×14×528	2	112	144	288	32	64	64	580K	119M
inception (4e)		14×14×832	2	256	160	320	32	128	128	840K	170M
max pool	3×3/2	7×7×832	0								
inception (5a)		7×7×832	2	256	160	320	32	128	128	1072K	54M
inception (5b)		7×7×1024	2	384	192	384	48	128	128	1388K	71M
avg pool	7×7/1	1×1×1024	0								
dropout (40%)		1×1×1024	0								
linear		1×1×1000	1							1000K	1M
softmax		1×1×1000	0								

[Szegedy et al. 2014]

São camadas de convolução, em seguida existem vários blocos de inception modules seguidos de camadas de Max Pooling que afetam as dimensões espaciais. Ao final, ao invés de uma Fully-connected layer, a GoogLeNet apresenta um passo intermediário com a camada de Global average pooling. Com isso, a taxa de acerto da CNN foi incrementada em 0, 6% [Szegedy et al. 2014].

5.5. AVALIAÇÃO DO DESEMPENHO DA REDE

Para a avaliação do desempenho do classificador foram utilizadas as métricas de acurácia, precisão, recall e F1-Score.

A acurácia mede à proporção das amostras rotuladas corretamente em relação ao conjunto total de amostras.

$$Acurácia = (TP + TN)/(TP + FP + FN + TN) \qquad (7)$$

- A precisão mostra dentre todas as classificações de classe Positivo que o modelo fez, quantas estão corretas.

$$Precisão = TP/(TP + FP) \quad (8)$$

- O Recall indica dentre todas as situações de classe Positivo como valor esperado, quantas estão corretas.

$$Recall = TP/(TP + FN) \quad (9)$$

- F1-Score é a média harmônica entre a precisão e recall, relacionando assim as duas variáveis.

$$F1Score = 2 \times (Recall \times Precisão)/(Recall + Precisão) \quad (10)$$

onde TP (True Positive) são os casos onde a classe é verdadeira e o classificador a classificou como verdadeira. TN (True Negative) são os casos onde a classe é falsa e o classificador a classificou como falsa. FP (False Positive) ocorre quando a classe é falsa e o classificador a classifica como verdadeira, e FN (False Negative) ocorre quando a classe é verdadeira mas o classificador a classifica como falsa.

Além das métricas descritas para avaliar os resultados dos classificadores, outra medida se fez necessária, visto que a NBR 15290 limita em um tempo máximo a tarefa de transcrição de programas ao vivo, ou seja, o atraso entre a entrada do sinal de voz e a escrita do texto não deve ser maior do que 4 segundos. Para avaliar essa característica, um teste de carga foi executado com 100 entradas das imagens das STFTs, e os tempos de cada etapa de processamento foram medidos.

O trabalho foi executado no ambiente de programação do MatLab 2018b, em um computador com sistema operacional Windows 10 Pro x64, tendo uma configuração composta por um processador Intel Pentium Gold G5400, com oito gigabytes de memória RAM, e placa de vídeo NVIDIA 1070, com quatro gigabytes de memória GDDR5.

5.6. COMPARAÇÃO COM MODELO DE REFERÊNCIA

Para efeito de comparação dos resultados obtidos com a rede GoogLeNet treinada com os espectrogramas das STFTs dos sinais de áudio, uma segunda rede com as mesmas características da primeira foi treinada com as imagens dos ceptrum das MFCCs extraídas dos mesmos sinais de áudio. As MFCCs foram escolhidas como modelo de referência por ser uma técnica consolidada, visto que suas etapas são modeladas para aproximar o sistema da forma com que o ouvido humano funciona. Todas as configurações aplicadas para o treinamento da rede treinada com as STFTs foram aplicadas ao treinamento da rede treinada com as MFCCs. Ambos os conjuntos de imagens (STFTs e MFCCs) possueam a mesma resolução, 224 $\times$ 224 $\times$ 3.

6. RESULTADOS E DISCUSSÃO

Os resultados foram obtidos em termos do tempo de transcrição e do desempenho na classificação das imagens dos espectrogramas das STFTFs e das imagens dos Cepstrum das MFCCs da rede GoogLeNet.

6.1. TEMPO DE TRANSCRIÇÃO

A Tabela 3 mostra os resultados dos tempo de processamento das etapas dos sinais de áudio com as imagens das STFTs, e a Tabela 4, mostra os correspondentes resultados obtido pelo método de referência, com a rede treinada com as imagens obtidas pelas MFCCs.

Para a etapa de remoção de silêncio, o tempo médio foi de 0, 12ms. O valor foi menor do que o obtido pelo modelo de referência. Apesar da função utilizada e dos arquivos de entrada serem os mesmos, variações no sistema operacional podem ter impactado nos resultados.

Para a extração das STFTs e construção do espectrogramas, em média, gastou-se 1, 00ms. Comparando com o tempo para a obtenção dos cepstrums das MFCCs, o processo de criar o espectrograma foi muito mais lento. Avaliando as etapas internas dessa função, nota-se que a construção da superfície que relaciona tempo, frequência e magnitude foi custoso para o sistema. A etapa de classificação para o modelo com a STFT levou em média 5, 32ms para classificar o áudio de entrada, tempo inferior aos 5, 92ms necessários para o modelo com as MFCC. Apesar de terem estruturas idênticas, o menor tempo de transcrição foi obtido pela rede treinada com as STFTs (6,43ms), compado com o tempo de transcrição obtido pela rede treinada com as imagens das MFCCs (6,46ms).

Tabela 3. Tempo (ms) de execução do modelo baseado nas STFTs.

	Remover silêncio	Extração STFT	Classificar	TOTAL
Máximo (ms)	0,18	1,40	6,36	7,94
Mínimo (ms)	0,11	0,96	5,16	6,22
Média (ms)	0,12	1,00	5,32	6,43

Tabela 4. Tempo (ms) de execução do modelo de referência - MFCCs.

	Remover silêncio	Extração MFCC	Classificar	TOTAL
Máximo (ms)	0,22	0,70	8,26	9,19
Mínimo (ms)	0,134	0,31	5,34	5,78
Média (ms)	0,16	0,38	5,92	6,46

6.2. RESULTADOS DA CLASSIFICAÇÃO

A Figura 6 mostra a acertividade por classe para o classificador treinado com as STFTs laranja), após oito épocas de treinamento, em um conjunto de teste com 30% da amostra.

Percebe-se que o resultado obtido pelas STFTs foi superior para todas as classes quando comparado ao resultado das MFCCs. Em 13 das trinta classes, o classificador obteve taxa de acerto igual a 100%. Em outras quatorze classes, a acertividade ficou acima dos 99%.

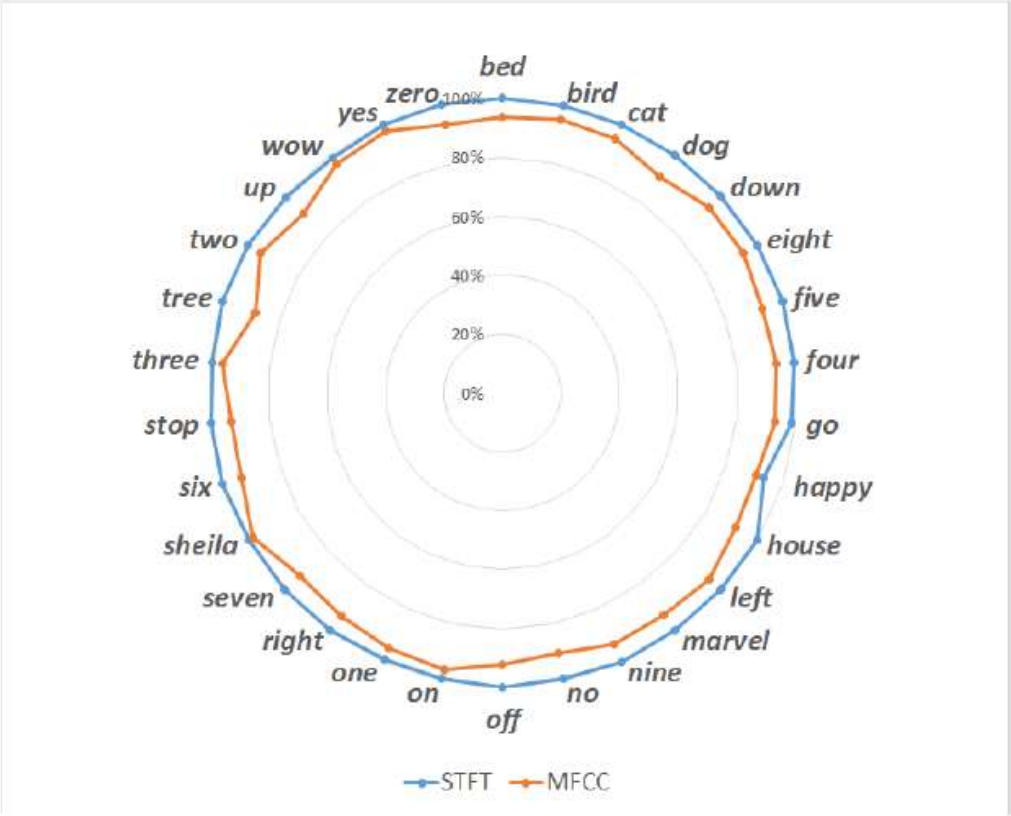
Apenas três classes tiveram resultados abaixo dos 99%, com destaque negativo para a classe happy, com 9% de taxa de erro.

O resultado de acurácia de cada classe indica que o classificador baseado nas STFTs superou o resultado do modelo de referência construído com as MFCCs. Como os dados estão desbalanceados, a acurácia não deve ser utilizada isoladamente para a avaliação do desempenho dos modelos [Sun et al. 2009], visto que é possível que a acurácia de uma única e exclusiva classe com maior peso possa estar aumentando e, com isso, melhorando a acurácia geral, mesmo que para as demais classes o resultado se mantenha ou piore.

Nessas situações onde o conjunto de dados é desbalanceado, é recomendável avaliar os resultados utilizando-se do F1-Score. Dessa forma, o gráfico contido na Figura 7 traz o resultado dos dois classificadores para as métricas, F1-Score, precisão, recall e acurácia.

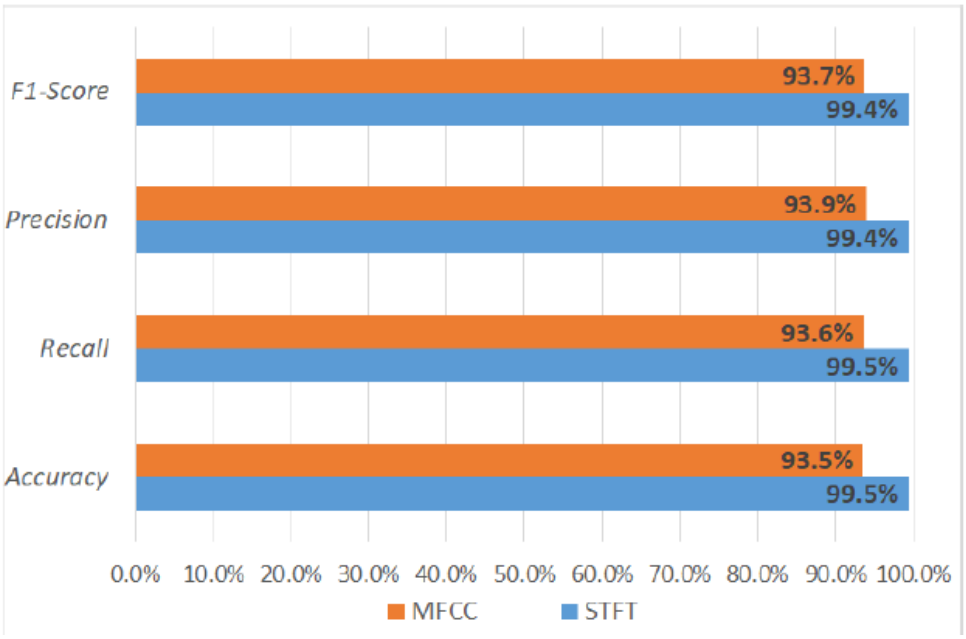
Em todos os parâmetros medidos, a abordagem utilizando as STFTs mostrou-se superior ao das MFCCs, tendo uma acurácia média de 99,5% e sem indicativos de over

Figura 6. Comparação das acurácias por classe dos modelos com STFT e de referência (MFCC)



Elaborada pelo autor.

Figura 7. Comparação entre as abordagens das MFCCs e STFTs sobre F1-Score, Precisao, recall e acurácia



Elaborada pelo autor.

fitting, visto que os resultados de F1-Score (99,4%), precisão (99,4%) e recall (99,5%) acompanharam a acurácia.

7. CONCLUSÃO

Nesse trabalho buscou-se uma solução para melhorar a qualidade da acessibilidade das legendas ocultas. Hoje, apesar dos melhores sistemas disponíveis no mercado atenderem ao critério de WER para a tarefa de transcrição de fala para texto, eles não conseguem obter resultados satisfatórios para atender a metodologia de medição NER, sendo essa a que está em vigor na legislação brasileira. O fato dos sistemas atuais utilizarem-se de modelos de linguagem e de análise de contexto para a tarefa de transcrição, permite que algumas palavras ao longo da pronúncia de uma sentença seja modificada, o que dificulta a inteligibilidade. Essas trocas de palavras são fatores que impactam no resultado NER.

Buscando uma solução para o problema discutido, esse estudo propôs uma abordagem de ASR que fosse capaz de transcrever fala em texto sem utilizar métodos auxiliares que possam reduzir a inteligibilidade, ao mesmo tempo, que respeitasse a NER e o tempo de transcrição definido pela NBR 15.290. A abordagem proposta consistiu na utilização das imagens dos espectros das STFTs e sua classificação pela rede neural GoogLeNet.

Para fins de comparação um método de referência que utilizou os cepstrums das MFCCs foi implementado e as imagens foram classificadas pela mesma rede utilizada para a classificação das imagens da STFTs. Os dados utilizados são arquivos de áudios de pronúncia de palavras sinteticamente criadas, no idioma inglês. Ao todo, o dataset apresentou 30 classes, desbalanceadas, com duração total de cada arquivo em um segundo.

O fato da implementação via STFTs ter superado o caminho tradicional das MFCCs pode estar relacionado a abordagem na classificação. Uma das principais vantagens da MFCC é sua capacidade de representar um número muito elevado de informações em poucos descritores, o que reduz o volume de dados e acelera o trabalho de treinamento e classificação. Entretanto, quando transformamos esses descritores em uma imagem, essa vantagem é perdida e uma desvantagem na aplicação da MFCC se destaca: a consolidação da informação do áudio em um conjunto de descritores faz com que haja perda de características do sinal original.

O resultado melhor na abordagem via STFT corrobora esse raciocínio, visto que na construção do espectrograma quase não há perda de informação do sinal de fala e, portanto, um maior número de informações relevantes são passadas ao classificador. Essa conclusão só foi possível devido a forma

idêntica ás quais as etapas inicias foram implementadas e também ao uso do mesmo classificador tanto para as MFCCs quanto para as STFTs. Todavia, os resultados obtidos mostraram que a abordagem do problema de reconhecimento de fala utilizando-se da capacidade das CNNs em classificar imagens, é um caminho de pesquisa para esse campo da ciência[Krizhevsky et al. 2012b].

Sobre os objetivos propostos, podemos afirmar que a abordagem de reconhecimento automático de fala a partir de imagens e com o uso das CNNs mostrou-se apta para a tarefa de transcrição de fala em texto, respeitando as definições da NBR 15.290. Para o caso da aplicação via MFCCs, o valor de NER ficou abaixo do permitido, 93, 54% contra 95%, mas para o tempo de transcrição o resultado superou o determinado pela normativa com ampla vantagem, 6, 46ms contra 4 segundos. Para a abordagem via STFT, os dois critérios foram alcançados. O valor de NER superou o estabelecido, 99, 48% contra 95%, e o tempo de transcrição, 6, 43ms contra 4 segundos.

REFERÊNCIAS

ABNT, N. (2016). 15290, acessibilidade em comunicação na televisão. Associação Brasileira de Normas técnicas, pages 15290–25.

Aleksic, P., Ghodsi, M., Michaely, A., Allauzen, C., Hall, K., Roark, B., Rybach, D., and Moreno, P. (2015). Bringing contextual information to google speech recognition. In Sixteenth Annual Conference of the International Speech Communication Association.

Buchner, J. (2017). Synthetic speech commands: A public dataset for single-word speech recognition.

Busson, A. J. G., Figueiredo, L. C., Santos, G. P. d., Damasceno, A. L. d. B., Colcher, S., and Milidiu, R. L. (2018). Desenvolvendo modelos de deep learning para aplicações multimídia no tensorflow. In Anais do XXIV Simpósio Brasileiro de Sistemas Multimídia e Web: Minicursos.

Crystal, D. (1988). Dicionário de Linguística e Fonética. Jorge Zahar Editor.

Das, A. (2015). Guide to signals and patterns in image processing: foundations, methods and applications. Springer.

Dave, N. (2013). Feature extraction methods lpc, plp and mfcc in speech recognition.

International journal for advance research in engineering and technology, 1(6):1–4.

Girshick, R. B., Donahue, J., Darrell, T., and Malik, J. (2013). Rich feature hierarchies for accurate object detection and semantic segmentation. CoRR, abs/1311.2524.

Goldberg, R. and Riek, L. (2000). A practical handbook of speech coders. CRC press.

Hermansky, H. (1990). Perceptual linear predictive (PLP) analysis of speech. J. Acoust. Soc. Am., 57(4).

Hirsch, H.-G. and Pearce, D. (2000). The aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditions. In ASR2000- Automatic Speech Recognition: Challenges for the new Millenium ISCA Tutorial and Research Workshop (ITRW).

Hu, J., Shen, L., and Sun, G. (2018). Squeeze-and-excitation networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 7132–7141.

Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37, pages 448–456.

Kaiming He, Xiangyu Zhang, S. R. J. S. (2015). Deep residual learning for image recognition. CoRR.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012a). Imagenet classification with deep convolutional neural networks. In Pereira, F., Burges, C. J. C., Bottou, L., and

Weinberger, K. Q., editors, Advances in Neural Information Processing Systems 25, pages 1097–1105. Curran Associates, Inc.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012b). Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems, pages 1097–1105.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2017). Imagenet classification with deep convolutional neural networks. Communications of the ACM, 60(6):84–90.

Lin, M., Chen, Q., and Yan, S. (2013). Network in network. arXiv preprint ar- Xiv:1312.4400.

Liu, J., Gao, X., Bao, N., Tang, J., andWu, G. (2017). Deep convolutional neural networks for pedestrian detection with skip pooling. pages 2056–2063. IEEE.

Paliwal, K. K. and Alsteris, L. D. (2005). On the usefulness of stft phase spectrum in human listening tests. Speech communication, 45(2):153–170.

Papandreou-Suppappola, A. (2018). Applications in time-frequency signal processing. CRC press.

Poularikas, A. D. (2010). Transforms and applications handbook. CRC press.

Robinson, A. (1997). British english example pronunciation dictionary (beep). Cambridge University, Cambridge.

Sanjaya,W. S. M., Anggraeni, D., and Santika, I. (2018). Speech recognition using linear predictive coding (lpc) and adaptive neuro-fuzzy (anfis) to control 5 dof arm robot. Journal of Physics: Conference Series, 1090.

Simonyan, K. and Andrew, Z. (2015). Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556, 6.

Sokolova, M. and Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. Information processing & management, 45(4):427–437.

Sun, Y., Wong, A. K., and Kamel, M. S. (2009). Classification of imbalanced data:

A review. International Journal of Pattern Recognition and Artificial Intelligence, 23(04):687–719. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S. E., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2014). Going deeper with convolutions. CoRR, abs/1409.4842.

Tusked, Z., Drepper, F. R., and Schl¨uter, R. (2012). Non-stationary signal processing and its application in speech recognition. In SAPA@INTERSPEECH.

Wisdom, S., Powers, T., Atlas, L., and Pitton, J. (2015). Enhancement and recognition of reverberant and noisy speech by extending its coherence. arXiv preprint ar-Xiv:1509.00533.

Xie, S., Girshick, R., Doll´ar, P., Tu, Z., and He, K. (2017). Aggregated residual transformations for deep neural networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 1492–1500.

Yuegang, W., Shao, J., and Hongtao, X. (2007). Non-stationary signals processing based on stft. In 2007 8th International Conference on Electronic Measurement and Instruments, pages 3–301. IEEE.

Zeiler, M. D. and Fergus, R. (2013). Visualizing and understanding convolutional networks. arXiv:1311.2901, 3.