CDIF 2023

2023

1º SEMINÁRIO DE CIÊNCIA DE DADOS DO IFSP

ISBN: 978-65-995529-1-5
CBL
9 786599 552915

INSTITUTO FEDERAL
São Paulo
Câmpus Campinas

INSTITUTO FEDERAL DE EDUCAÇÃO CIÊNCIA E TECNOLOGIA DE SÃO PAULO - CAMPUS CAMPINAS

I SEMINÁRIO DE CIÊNCIA DE DADOS DO IFSP (CDIF)
ANAIS

1ª Edição
30 de março de 2023

Comissão Organizadora
Prof. Dr. Samuel Botter Martins (Coordenador Geral)
Prof. Dr. Andreiwid Sheffer Corrêa
Profa. Dra. Bianca Maria Pedrosa
Profa. Dra. Eliana Alves Moreira
Prof. Me. Everton Josué da Silva
Prof. Me. Fábio Feliciano de Oliveira

IFSP Câmpus Campinas
Rua Heitor Lacerda Guedes, 1000 – Cidade Satélite Íris – Campinas-SP

Dados Internacionais de Catalogação na Publicação (CIP)
(Câmara Brasileira do Livro, SP, Brasil)

Seminário de Ciências de Dados do IFSP (1. : 2023 : São Paulo, SP)
Anais 1º CDIF 2023 [livro eletrônico] / organização Andreiwid Sheffer Corrêa...[et al.] ; coordenação Samuel Botter Martins. -- 1. ed. -- Campinas, SP : Instituto Federal de Educação, Ciência e Tecnologia de São Paulo, Câmpus Campinas, 2023.
PDF

Vários autores.
Outros organizadores: Bianca Maria Pedrosa, Eliana Alves Moreira, Everton Josué da Silva, Fábio Feliciano de Oliveira.
Bibliografia.
ISBN 978-65-995529-1-5

1. Banco de dados - Desenvolvimento 2. Ciência da computação 3. Inteligência artificial 4. Pesquisa científica I. Corrêa, Andreiwid Sheffer. II. Pedrosa, Bianca Maria. III. Moreira, Eliana Alves. IV. Silva, Everton Josué da. V. Oliveira, Fábio Feliciano de. VI. Martins, Samuel Botter.

23-158077 CDD-005.73

Índices para catálogo sistemático:

1. Dados : Estruturas : Processamento de dados 005.73

Aline Graziele Benitez - Bibliotecária - CRB-1/3129

Prefácio

É com grande prazer que apresentamos os anais do 1º Seminário de Ciência de Dados do IFSP (CDIF), realizado em 30 de março de 2023 no auditório do IFSP câmpus Campinas. Este evento reuniu pesquisadores, profissionais e estudantes de graduação e pós-graduação de computação para discutir e compartilhar conhecimentos sobre a ciência de dados e suas aplicações.

O seminário contou com uma palestra sobre o uso de ciência de dados na área de saúde e demografia, além de apresentações de trabalhos originais realizados por estudantes do curso de *Especialização em Ciência de Dados* ofertado no mesmo câmpus. Os trabalhos apresentados abordaram temas diversos, como análise de sentimentos em redes sociais, classificação de imagens médicas, predição de valores de ações em bolsa de valores, entre outros. Esses trabalhos mostram a diversidade e a relevância das pesquisas realizadas pelos estudantes de pós-graduação do IFSP Campinas.

Gostaríamos de agradecer aos organizadores e apoiadores do seminário que tornaram possível a realização deste evento. Também gostaríamos de agradecer aos palestrantes e autores dos trabalhos apresentados, que contribuíram para o sucesso do evento.

Esperamos que estes anais possam servir como uma fonte de inspiração e referência para os estudantes e profissionais que desejam aprofundar seus conhecimentos em ciência de dados. A ciência de dados é uma área em constante evolução e tem se mostrado cada vez mais relevante em diversos setores, como saúde, finanças, varejo, entre outros.

Por fim, gostaríamos de ressaltar a importância de eventos como este para promover o diálogo e o intercâmbio de conhecimentos entre os diferentes atores envolvidos na ciência de dados. Esperamos que este seja apenas o primeiro de muitos outros seminários em nosso câmpus dedicados a essa área fascinante e em constante evolução.

Palestra

Título: Demografia, saúde e ciência de dados.
Palestrantes:
Profa. Dra. Luciana Correia Alves
Prof. Me. Carlos Eduardo Beluzo

Artigos

Identificação de discurso de ódio contra asiáticos no Twitter durante a pandemia

Priscila Marques de Oliveira
Instituto Federal de Educação, Ciência e Tecnologia de São Paulo - Campus:Campinas.
Campinas, Brasil
p.marques@aluno.ifsp.edu.br

Ricardo Barz Sovat
Instituto Federal de Educação, Ciência e Tecnologia de São Paulo - Campus:Campinas.
Campinas, Brasil
sovat@ifsp.edu.br

***Resumo* — Com a chegada da pandemia do covid-19 a internet se tornou um meio de comunicação ainda mais utilizado, já que os *lockdowns* eram medidas adotadas na tentativa de conter a propagação da doença. Com o acontecimento do primeiro caso na cidade de Wuhan, China, surgiram nas redes sociais manifestações de ódio direcionadas especialmente a pessoas de ascendência asiática. Devido à quantidade e à força que estas manifestações ganharam, elas saíram do âmbito virtual e atingiram o mundo real, resultando não só em violência verbal, mas muitas vezes física. O Brasil abriga a maior comunidade japonesa fora do Japão. Além de japoneses há outras comunidades asiáticas no Brasil. Por ser um país grande e com acesso aos meios digitais pela maioria da população, casos de manifestação de ódio podem ocorrer e levar a manifestações no mundo real. Devido a estes fatos, utilizando o twitter como principal fonte de dados, buscou-se verificar a existência de discurso de ódio também em Português. Para esta tarefa utilizamos como estratégia de treinamento o BERT, um *transformer* desenvolvido pelo Google para o inglês, idioma mais usado mundialmente. Depois, com as técnicas de *fine tuning* e *transfer learning*, treinamos um modelo que identifica o discurso de ódio em Português. O código e os dados estão disponíveis em https://github.com/prypmo/HateSpeech.**

Palavras-chave — PLN, Discurso de ódio, Covid-19, BERT, Transfer Learning, Adapter, AdapterHub

I. Introdução

As redes sociais se tornaram um meio de comunicação muito representativo nos dias atuais. Por meio delas, muitas discussões em vários âmbitos têm acontecido. Por ser um meio acessível a toda a população é possível encontrar diversas formas de pensamento e expressão de sentimentos, que muitas vezes não são agradáveis. Em determinados momentos podemos encontrar nas redes sociais algo conhecido como discurso de ódio, em inglês, *hate speech*, que pode ser definido como o uso de linguagem agressiva que deprecie um grupo ou pessoa, baseado em características como raça, cor, etnia, gênero, religião ou outros tipos de características [1].

Desde a notificação do primeiro caso de COVID-19 na cidade de Wuhan, China [2], a comunidade asiática tem sido alvo de manifestações de ódio, online e offline. Estes assédios em redes sociais, além de incitar ataques virtuais, também incentivaram ataques físicos e violentos. Nos Estados Unidos, dentre as denúncias feitas por asiáticos, humilhações verbais representam 68,1%, xingamentos 20.5% e agressões físicas, 11.1% [3][4].

Por este motivo, estudos para a detecção, no Twitter principalmente, destes ataques à comunidade asiática durante a pandemia têm sido conduzidos [5][6]. Em sua maioria estes estudos possuem como idioma alvo para estudo o inglês [5][6]. No Brasil, onde a maior colônia japonesa está localizada [7], estes ataques físicos e verbais também têm ocorrido [8][9]. Contudo, não há nenhum estudo de identificação utilizando o português como idioma padrão.

Assim, este trabalho almeja construir um modelo baseado em Ciência de Dados que identifique um discurso de ódio ao ler mensagens do Twitter.

Como proposta para este trabalho, empregaram-se técnicas de de PLN (Processamento de Linguagem Natural) para identificação desse tipo de conteúdo agressivo, utilizando como idioma padrão o português. Juntamente com isso utilizou-se o BERT como estratégia de treinamento e técnicas de *fine tuning* e *transfer learning*, para gerar um modelo que pudesse executar a tarefa descrita.

II. Database

A. Coletando Dados

Para o desenvolvimento deste trabalho não houve possibilidade de utilizar bases de *tweets* já conhecidas e devidamente anotadas. Os trabalhos utilizados como base[5][6] fornecem dados, porém todos em inglês. Desta forma, utilizando a ferramenta SNScrape, para a coleta de dados, foram coletados dados de março de 2020 a junho de 2020, período entre a expansão da contaminação e a primeira onda da pandemia no Brasil [10]. Para filtrar a pesquisa de tweets relacionados à pandemia, utilizamos para a busca as palavras chaves "coronavírus", “covid”, “morcego”, “china”, “pastel de flango”, “chineses”, “virus chines”, “peste chinesa”, “xingling” e “xing ling”. Estes termos foram utilizados porque sua maioria faz referência à doença ou já são termos conhecidos no idioma português para se referir de forma pejorativa a comunidade asiática.

B. Preparo da Base

Após a obtenção dos dados, foi verificada uma discrepância entre a quantidade de tweets por dia. Sendo assim e levando-se em conta o tempo disponível para a anotação, foram selecionados 200 por dia, de forma

aleatória. Esta aleatoriedade procurou manter uma quantidade de tweets suficiente para o treinamento ao mesmo tempo que não introduzisse nenhum viés temporal na amostra. Os dados foram anotados, utilizando-se como regra que haveria apenas duas possibilidades, discurso de ódio ou não. Esta classificação foi feita manualmente sob a ótica apresentada em [1] que busca enquadrar discursos de ódio em mensagens que possam ser consideradas hostis ou abusivas. Não necessariamente todos os tweets filtrados pela busca dos termos listados na subseção anterior estavam presentes em mensagens anotadas como "*hate speech*". Deve-se chamar a atenção para a existência de um grau de subjetividade neste ponto.

III. TRANSFORMERS

Dá-se o nome de *transformer* a uma arquitetura de redes neurais proposta pela Google LLC em 2017[11]. O tipo de problema a que esta arquitetura se destinava envolvia o reconhecimento de padrões apresentados na forma de sequências de valores, sobretudo sequências longas. Anteriormente, esta área de aplicação era dominada por modelos baseados em RNN (*recurrent neural networks* – redes neurais recorrentes). A abordagem dos transformers baseia-se principalmente no conceito de auto-atenção, o que permite que essas longas sequências dispensem tanto o necessário alinhamento dos valores para RNNs quanto o emprego de convolução, este central ao Deep Learning. A informação a ser aprendida é extraída da sequência como um todo, o que facilita manter o contexto. A PLN foi muito beneficiada pelo advento dos transformers.

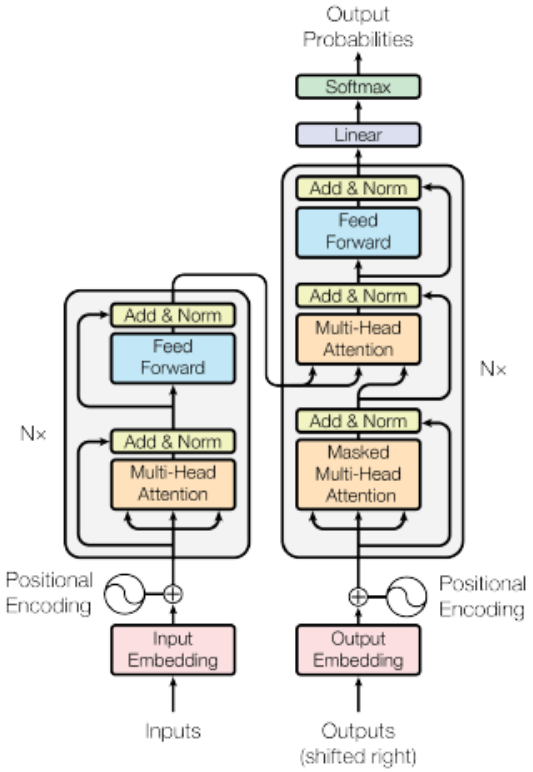


Figura 1. Arquitetura de transformers. Figura extraída de [11]

Ao mesmo tempo, o emprego de datasets mais reduzidos começou a ser possível pelo uso de um método de aprendizado chamado Transfer Learning.

A união dessas técnicas aumentou a eficiência do tratamento de longas sequências e é responsável por dois dos mais difundidos transformers sendo empregados no momento: o GPT (*Generative Pretrained Transformer*) e o BERT (*Bidirectional Encoder Representations from Transformers*). Este último e suas variações foram utilizados neste trabalho.

A. *BERT*

BERT, **B**idirectional **E**ncoder **R**epresentations from **T**ransformers, é um transformer desenvolvido pelo Google [12] que atinge o estado da arte em Processamento de Linguagem Natural. BERT foi treinado utilizando dados do Wikipedia e Google Book Corps. Diferente de alguns modelos existentes como ELMo [13], onde a leitura da sentença era feita da direita para esquerda, BERT pode fazer a leitura nos dois sentidos, esquerda para direita e direita para esquerda ao mesmo tempo. A partir deste modo de leitura, BERT utiliza duas técnicas de PLN, o MLM ou "*Masked Language Model*" e "*Next Sentence Prediction*". O MLM substitui uma palavra da sentença por uma máscara e a partir do contexto efetua a predição. Já o *Next Sequence Prediction* verifica a relação entre duas sentenças. Esse processo é feito ao mesmo tempo. Estes processos só são possíveis devido à utilização de transformers no BERT.

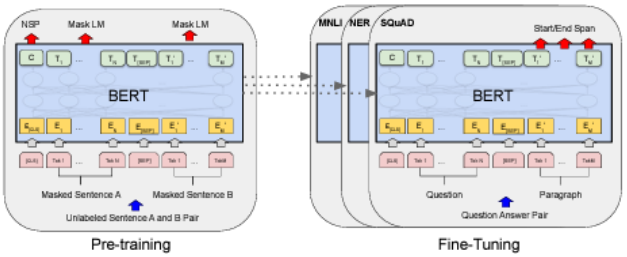


Figura 2. Arquitetura do BERT. Figura extraída de [12]

B. *XLM-RoBERTa*

RoBERTa [14] é uma variação do BERT desenvolvida pela então Facebook Inc.. Sua implementação é basicamente a mesma, porém trabalha com batches e taxas de aprendizado maiores durante o treinamento. A etapa de *Next Sentence Prediction* também é removida. Existe também o XLM-Roberta [15], que se trata de uma variação do RoBERTa,com suporte a outros idiomas além do inglês.

C. *DistilBERT*

DistilBert [16] é uma versão reduzida de BERT, que emprega a técnica de *Knowledge Distillation*[17], também conhecida como aprendizado professor-aluno. Nela, um modelo de dimensão menor é treinado, visando reproduzir o comportamento do modelo maior. Distilbert, tem seu tamanho reduzido 40% em relação ao BERT, sendo 60% mais rápido e mantendo as capacidades de aprendizado de linguagem de 97% em relação ao BERT.

IV. TRANSFER LEARNING

No mundo real, se soubermos como é uma maçã, sua forma e cor, podemos através desse conhecimento reconhecer peras. Se aprendermos como tocar um órgão, podemos facilmente aprender como tocar um teclado ou

piano. A partir destes exemplos, podemos verificar que é possível perceber que um conhecimento específico pode ser utilizado em diferentes tarefas que possuem alguma conexão, peras e maçãs são frutas, assim como teclado, órgão e piano são instrumentos musicais. Pensando em Machine Learning, essa transferência de conhecimento pode ser muito útil. Dado que se tem um modelo que já foi treinado em determinada tarefa, podemos utilizar esse conhecimento para aplicar em uma tarefa diferente. A definição para Transfer Learning, segundo [18], diz que o objetivo do transfer learning é melhorar a predição de uma uma tarefa de PLN T_t com um grupo de dados não treinado D_t utilizando o conhecimento adquirido a partir de um grupo de dados treinados D_s e uma tarefa de PLN T_s já predita. Nesta definição pode ser entendido como T_t tarefa-alvo e D_t como domínio-alvo e T_s como tarefa-origem e D_s dominio-origem.

Transfer Learning pode ser subdividido em três categorias: Transductive Transfer Learning, Inductive Transfer Learning e Multi-Tasks Transfer Learning, que serão descritas abaixo, de acordo com [19].

A. *Transductive Transfer Learning*

Quando a tarefa de PLN é a mesma para a origem e o alvo, mas os domínios são diferentes e a tarefa-alvo não possui (ou possui pouquíssimos) dados identificados, é definido como Transductive Transfer Learning. As duas subcategorias que podemos associar à transductive são:

1) Domain Adaptation: Se refere ao processo de adaptar o aprendizado a um novo domínio, diferente do domínio original. É muito útil quando os dados identificados do domínio-alvo são escassos.

2) Cross-lingual Learning: Neste caso, seria a adaptação para um domínio-alvo de idioma diferente. Ainda falando sobre análise de sentimentos, podemos utilizar como exemplo, tweets em inglês (um idioma comum, que possui vários dados identificados) para tweets em árabe, que não possui muitas bases identificadas.

B. *Inductive Transfer Learning*

É identificada quando as tarefas de PLN são diferentes entre origem e alvo e apenas o domínio-alvo possui dados identificados. Este tipo de técnica se adequa muito bem aos modelos pré-treinados como BERT e afins.

Inductive transfer learning pode ser dividido também em duas categorias, que por suas vez possuem subcategorias..

1) Multi-task Learning: Trata-se do processo de aprendizado de muitas tarefas PLN ao mesmo tempo. Usando como exemplo um modelo pré-treinado, efetuamos a transferência de conhecimento para várias tarefas de forma paralela.

2) Sequential Transfer Learning: A partir de um modelo pré treinado, o transfer learning é feito para várias tarefas, porém uma de cada vez. A cada passo, uma tarefa diferente é aprendida. Este modelo acaba sendo mais lento. Pode ser dividido em quatro categorias.

a) Fine-tuning: a partir de um modelo pré-treinado, a nova tarefa aprenderá uma função que mapeie os parâmetros do modelo pré treinado. A taxa de aprendizado pode ser diferente também podendo ser inseridos novos parâmetros a essa função.

b) Adapter modules: A partir de um modelo pré-treinado, são inseridas novas camadas com pesos inicializados aleatoriamente. Os parâmetros originais também são treinados, porém em menor quantidade.

c) *Feature based:* aqui o interesse é apenas em aprender sobre uma representação específica como caractere, palavra, sentença ou parágrafo embeddings.

d) Zero-shot: esta seria a mais simples das abordagens. Nenhuma alteração é feita no modelo pré-treinado ou em seus parâmetros, nada que otimize o processo.

V. Adapter Modules

Adapter modules, ou *adapter-based tuning* é uma abordagem de transfer learning que propõe a inserção de novas camadas à rede original[20]. São inicializados aleatoriamente novos pesos nas camadas de adapter e os pesos originais são mantidos sem alteração. Os parâmetros da rede original são congelados. Devido a isto estes parâmetros podem ser compartilhados entre outras tarefas.

Com a possibilidade de treinar várias tarefas de PLN sem a necessidade de re-treinar todos os parâmetros do modelo, seu uso é muito recomendado em contexto de cloud services.

A arquitetura dos adapters modules propõe a inserção de camadas no modelo original. Essas camadas têm seus pesos inicializados aleatoriamente. Os parâmetros já existentes são compartilhados entre as camadas, porém estão congelados, sem alteração. É introduzido apenas um pequeno número de parâmetros no modelo, para que a rede original se mantenha sem alterações e o treinamento seja estável. Os adapters são inicializados com uma função de ativação não-linear que influencia na distribuição das ativações.

Na figura 1, é possível verificar detalhadamente o funcionamento e arquitetura dos adapters.

Para o treinamento de menos parâmetros, adapters utilizam a seguinte arquitetura: as dimensões originais das features d são projetadas para uma dimensão menor, m. Uma função de ativação não linear é aplicada e o resultado projetado novamente para a dimensão d. A função $2md + d + m$ representa o total de parâmetros adicionados para cada camada, incluindo os vieses (biases).

A. *Resultados do Treinamento*

Para o treinamento e comparação utilizamos os modelos BERT, DistilBERT e RoBERTa XML. Para todos os modelos, os dados foram divididos entre validação treinamento e teste. Para o teste, foi separado 20% do valor total da base de dados. Para treinamento e validação foram separados, 80% e 20% respectivamente, do montante que sobrou. Para o tamanho de *batches* e quantidade de épocas, foram efetuados testes até atingir os resultados apresentados. O mesmo foi feito para a definição dos valores utilizados no otimizador Adam. Como métrica foi utilizada a de *Matthews Correlation Coefficient*, MCC. Esta métrica retorna um número decimal, que pode variar entre -1 e 1. Quanto mais o valor estiver próximo de 1, melhor a predição do modelo. Todos os modelos foram treinados

com o mínimo de otimizações em seus algoritmos e hiperparâmetros.

Assim, como esperado e demonstrado pelas explicações acima, existe uma diferença no desempenho de cada um. Podemos verificar estes resultados na tabela 1.

Para o treinamento utilizando adapter-based tuning, utilizamos a mesma base de códigos, mas agora com a camada extra de adapter. Para facilitar, utilizamos como framework o AdapterHub [21] que é fornecido pela Huggingface Inc.. Foi utilizada a mesma estrutura dos modelos anteriores, efetuando apenas algumas modificações em número de épocas e taxa de aprendizado. A métrica utilizada também foi a mesma, MCC. Foram treinados 1.487.427 parâmetros com a utilização de adapters, contra 167.356.416 que seriam treinados fazendo o fine-tuning convencional. O resultado de MCC obtido foi de 0.62, muito semelhante ao encontrado no treinamento do BERT.

Podemos verificar a comparação entre os modelos na tabela 2.

TABELA 1. COMPARAÇÃO DE RESULTADOS ENTRE OS MODELOS BERT, DISTILBERT E XLMROBERTA

Comparação de modelos e seus MCC			
MCC	BERT	DistilBERT	XLM RoBERTa
	0.62	0.61	0.64

TABELA 2. COMPARAÇÃO DE RESULTADOS ENTRE ADAPTERMODULES E FINETUNING

Comparação entre Fine-Tuning e Adapters Module		
	Quantidade de parâmetros	MCC
BERT	167.356.416	0.62
XLMRoBERTa	278.045.186	0.64
DistilBERT	135.326.210	0.61
AdapterModule (BERT)	1.487.427	0.62

Além deste treinamento, foi gerado um adapter que futuramente será disponibilizado no adapterhub.ml, para que outras pessoas possam utilizá-lo em seus experimentos. Este adapter será o primeiro para o idioma português e tarefa de análise de sentimento.

VI. CONCLUSÕES

Podemos verificar que os transformers são realmente ferramentas muito poderosas para tarefas de PLN. Além disso, a utilização de adapters aliada aos transformers, como uma alternativa ao transfer learning comum, nos abre novas perspectivas.

É interessante ressaltar o fato de ter sido criada e disponibilizada uma base de dados em Português anotada, em decorrência dos requisitos deste trabalho. Espera-se que ela se constitua em uma contribuição útil para a tarefa de Análise de Sentimento, permitindo que se escape dos trabalhos focados apenas no idioma inglês.

Como meta de trabalhos futuros, a ideia é que o adapter gerado neste trabalho possa ser disponibilizado, como uma alternativa em Português para a tarefa de Análise de Sentimentos.

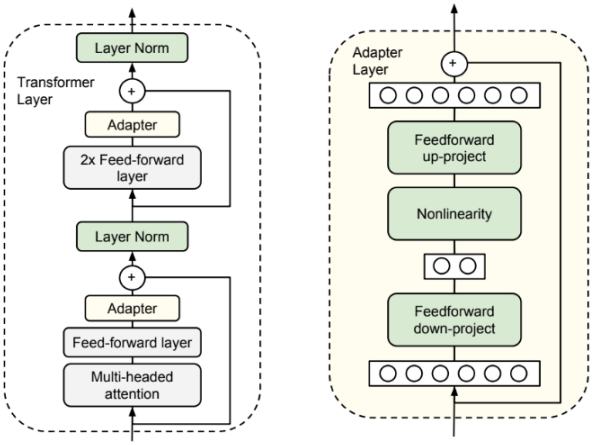


Figura 3: Arquitetura da utilização de adapter modules. Figura extraída de [20] .

REFERENCES

[1] Anna Schmidt and Michael Wiegand. 2017. "A survey on hate speech detection using natural language processing". Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media, April 3, 2017,

[2] Smriti Mallapaty. "After the WHO report: what's next in the search for COVID's origins." Nature. https://www.nature.com/articles/d41586-021-00877-4 (acesso em jun. 06, 2022)

[3] CSHE. "Fact sheet: Anti-Asian prejudice March 2021 by center for the study of hate & extremism.". www.csusb.edu/sites/default/files/FACT%20SHEET-%20Anti-Asian%20Hate%202020%20rev%203.21.21.pdf (acesso em maio 02, 2022)

[4] AAPI. "Stop aapi hate national report.". https://stopaapihate.org/2020-2021-national-report/(acesso em maio 02, 2022)

[5] Bing He, Caleb Ziems, Sandeep Soni, Naren Ramakrishnan, Diyi Yang, and Srijan Kumar. 2022. "Racism is a virus: anti-asian hate and counterspeech in social media during the COVID-19 crisis". In Proceedings of the 2021 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM '21). Association for Computing Machinery, New York, NY, USA, 90–94. https://doi.org/10.1145/3487351.3488324

[6] Bertie Vidgen, et al. "Detecting East Asian prejudice on social media". In Proceedings of the Fourth Workshop on Online Abuse and Harms, pages 162–172, Online. Association for Computational Linguistics.

[7] Cintia Cury. "Estado tem cerca de 1 milhão de japoneses e descendentes." Gov. Estado São Paulo. https://www.saopaulo.sp.gov.br/ultimas-noticias/estado-tem-cerca-de-1-milhao-de-japoneses-e-descendentes/ (acesso em maio 02, 2022)

[8] Tammaro, Rodrigo. "População de origem asiática é vítima de violência e preconceito na pandemia." Jornal USP, São Paulo, https://jornal.usp.br/atualidades/populacao-de-origem-asiatica-e-vitima-de-violencia-e-preconceito-na-pandemia/ (acesso em maio 02, 2022)

[9] Nakamura, J. Terao, S.0 "Brasileiros de ascendência asiática relatam ataques racistas durante a pandemia." Folha de São Paulo, https://www1.folha.uol.com.br/cotidiano/2020/05/brasileiros-de-ascendencia-asiatica-relatam-ataques-racistas-durante-a-pandemia.shtml (acesso em maio 02, 2022)

[10] Estrada, Camile Duque. Nóbrega, Lidiane. "Covid-19: balanço de dois anos da pandemia aponta vacinação como prioridade".FioCruz. https://www.fiocruzbrasilia.fiocruz.br/covid-19-balanco-de-dois-anos-da-pandemia-aponta-vacinacao-como-prioridade/ (acesso em jan. 12, 2023)

[11] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. “Attention is all you need”. In Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17). Curran Associates Inc., Red Hook, NY, USA, 6000–6010.

[12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

[13] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018.”Deep Contextualized Word Representations”. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

[14] Liu Zhuang, Lin Wayne, Shi Ya, and Zhao Jun. 2021. “A Robustly Optimized BERT Pre-training Approach with Post-training”. In Proceedings of the 20th Chinese National Conference on Computational Linguistics, pages 1218–1227, Huhhot, China. Chinese Information Processing Society of China.

[15] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. “Unsupervised Cross-lingual Representation Learning at Scale”. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 8440–8451, Online. Association for Computational Linguistics.

[16] Sanh, Victor, Debut, Lysandre, Chaumond, Julien and Wolf, Thomas. "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter." arXiv preprint arXiv:1910.01108 (2019).

[17] Hinton, Geoffrey and Vinyals, Oriol and Dean, Jeff. “Distilling the Knowledge in a Neural Network”. arXiv. https://doi.org/10.48550/arxiv.1503.02531

[18] S. J. Pan and Q. Yang, "A Survey on Transfer Learning," in IEEE Transactions on Knowledge and Data Engineering, vol. 22, no. 10, pp. 1345-1359, Oct. 2010, doi: 10.1109/TKDE.2009.191.

[19] Alyafeai, Zaid, Maged Saeed AlShaibani and Irfan Ahmad. “A Survey on Transfer Learning in Natural Language Processing.” ArXiv abs/2007.04239 (2020)

[20] Demi Guo, Alexander Rush, and Yoon Kim. 2021. “Parameter-Efficient Transfer Learning with Diff Pruning”. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 4884–4896, Online. Association for Computational Linguistics.

[21] Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulić, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. 2020. “AdapterHub: A Framework for Adapting Transformers ”. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pages 46–54, Online. Association for Computational Linguistics.

Detecção de Doenças em Folhas de Plantas usando Deep Learning

1st Amanda Rodrigues da Silva
Instituto Federal de Educação, Ciência e Tecnologia de São Paulo
Campinas, Brasil
rodrigues.amanda1@aluno.ifsp.edu.br

2nd Samuel Botter Martins
Instituto Federal de Educação, Ciência e Tecnologia de São Paulo
Campinas, Brasil
samuel.martins@ifsp.edu.br

Desde o início do século XXI, a agricultura tem atuado no que se conhece por "agricultura 4.0", um modelo de produção e gestão baseado em sistemas e dispositivos inteligentes. Apesar do crescimento de produção e valor agregado proporcionado por processos otimizados, o combate a doenças e pragas ainda é uma parcela dos gastos produtivos e, quando não se consegue identificar e combater a doença, a produção sofre perdas parciais ou totais. O presente estudo apresenta uma avaliação de modelos de Deep Learning (DL) para a detecção de doenças em folhas de plantas e, para isso, utiliza-se da base de dados pública PlantDoc, pré-processada com técnicas de data augmentation para a base de treino. Um modelo pré-treinado é utilizado como base e o método de transfer-Learning é aplicado considerando um classificador com hiperparâmetros da literatura. Os dois modelos são comparados considerando i) acurácia, ii) precisão e iii) tempo de execução. O modelo usando a rede VGG16-Sequential com método de transfer-Learning usando pesos treinados na base imagenet mostrou-se mais acurado e preciso tanto para a detecção de doenças quanto para a tentativa de diagnóstico das doenças, porém o tempo de execução deste é mais de 32 vezes maior que o do modelo usando CNN-Sequencial. Para a detecção de doenças obteve-se precisão de 0.71 e acurácia de 0.88 para o modelo composto pela CNN-Sequential, e precisão de 0.81 e acurácia de 0.91 para o modelo usando VGG16-Sequential.

Palavras Chave — Deep Learning, processamento de imagem, doença em planta, transfer-Learning

I. INTRODUÇÃO

A agricultura desempenha um papel fundamental na economia mundial e, com a expansão contínua da população humana, há, cada vez mais, uma pressão nos sistemas de produção agrícola, que, para serem capazes de atingir as expectativas e cobranças estabelecidas, precisam adotar métodos modernos e abordagens baseada em análise de dados de forma a otimizar sua produtividade enquanto mitiga os impactos ambientais causados em seus processos [7]. Além disso, a agricultura também é, muitas vezes, considerada uma prioridade de "segurança nacional" pelos países, pois esses produtos são necessários para existir, enquanto a maioria dos itens de manufatura não é tão essencial e ainda há de se considerar que muitas nações têm o agronegócio como fonte principal de receita [1].

Considerando o mercado brasileiro, um estudo publicado pelo Centro de Estudos Avançados em Economia Aplicada (CEPEA) em março deste ano [2] mostrou que o agronegócio foi responsável por mais de 27% do Produto Interno Bruto (PIB) brasileiro em 2021, com receita de aproximadamente R$ 183 bilhões e um crescimento de 8,36% comparado ao ano anterior. Considerando-se apenas a fração agrícola - em detrimento da pecuária - o PIB agregado do ano foi de R$ 243 bilhões, um aumento de 15,88% [2].

Apesar da relevância econômica do agronegócio para a economia mundial, a introdução à aplicação de técnicas de Deep Learning (DL) nesse setor e, em particular, ao campo de diagnóstico de doenças em plantas, começou a ser feita apenas na última década [15], já que, anteriormente, o assunto era estudado apenas considerando abordagens genômicas de identificação de genes e proteínas relevantes. Com o avanço da capacidade de processamento de dados e o desenvolvimento dos algoritmos de aprendizagem de máquina (do inglês, *Machine Learning - ML*), tornou-se possível o uso de métodos que automatizam o processo de construção de modelos que aprendem iterativamente com os dados para obter insights sem programação explícita [15]. Esses métodos tornam as análises mais poderosas e, além de ser uma ferramenta mais eficiente na identificação dos genes e proteínas, ainda permitem a realização do diagnóstico de plantas com doenças manifestas através do processamento e classificação de imagens [15].

A relevância da atuação na prevenção e no combate a doenças agrícolas dá-se por estas serem responsáveis por uma perda de 20 a 40% da produtividade mundial. Os autores ainda afirmam que as doenças de plantas, em particular, apesar de não serem as maiores responsáveis pelas maiores reduções de rendimento, têm impacto na qualidade da colheita e na segurança alimentar dos consumidores [9]. Dessa forma, o objetivo deste trabalho é realizar a detecção de doenças em folhas de plantas usando modelos de *Deep Learning*, considerando uma rede pré-treinada e uma sem pré-treinamento, além de técnicas de processamento e balanceamento dos dados para tornar possíveis as análises desejadas.

II. APLICAÇÕES DE DEEP LEARNING EM DETECÇÃO DE DOENÇAS EM FOLHAS DE PLANTAS

A comparação entre algoritmos de *Machine Learning* (ML) e *Deep Learning* (DL) para detecção de doenças em folhas de citrus foi feita comparando-se três algoritmos de cada abordagem e notou-se [13] que modelos de DL performam melhor para a detecção de doenças do que os modelos de ML. A Tabela 1 expõe os resultados apresentados em [13].

TABELA I ACURÁCIA DOS MODELOS DE ML E DP NA DETECÇÃO DE DOENÇAS NAS FOLHAS DE CITRUS [13]

Tipo	*Algoritmo*	*Acurácia*
ML	Support Vector Machine (SVM)	87%
ML	Random Forest (RF)	76.8%
ML	Stochastic Gradient Descent (SGD)	86.5%
DL	Inception-v3	89%
DL	VGG-16	89.5%
DL	VGG-19	87.4%

Fonte: [13]

A Tabela 2 mostra o estado da arte da aplicação de redes neurais convolucionais (CNN, *convolutional neural*

network) para a detecção de doenças e 15 diferentes estudos foram mostrados, todos com acurácia superior a 80% para as bases de imagens, conforme apresentado em [4].

TABELA II ESTADO DA ARTE DE REDES NEURAIS CONVOLUCIONAIS (CNNS) PARA A DETECÇÃO DE DOENÇAS EM PLANTAS [4]

Autor	*Método*	*Dataset*	*Classes*	*Acurácia*
Mohanty et al.[17]	AlexNet e GoogleNet	PlantVillage	38	99.27% 99.34%
Ramcharan et al. [18].	Inception V3 based on GoogleNet	Cassava dataset	6	93%
Fuentes et al.[19]	Faster R-CNN, R-FCN,, SSD combined with VGG-16 and ResNet	custom Tomato Diseases and Pests Dataset 5000 imagens	9	83%
Pawara et al.[20]	AlexNet e GoogleNet	AgriPlant Dataset LeafSnap Dataset Folio Dataset	10 184 32	96.37% 98.33% 89.51% 97.66% 97.67% 97.63%
Ferentinos et al. [21]	AlexNetOWTBn e VGG	custom dataset 87848 images	58	99.49%, 99.53%
Ramachara n el al. [22]	MobileNet- SSD	Cassava dataset	6	80.6%
Geetharamani et al. [23]	CNN	PlantVillage with data augmentation	39	96.46%
Chen et al. [24]	VGG-19 pre-trained on ImageNet with Inception module	Maize PlantVillage	4	92%
Chen et al. [25]	DenseNet	Maize PlantVillage	4	98.50%
Chen et al. [26]	MobileNet-V2	PlantVillage	38	99.71%
Chen et al. [27]	DenseNet	custom dataset 1000 images	5	97.60%
Li et al. [28]	CNN	NBAIR Li et al. dataset	50 10	95.40% 96.20%
Chen et al. [29]	MobileNet-V2 e Attention Mechanism along with a Classification Activation Map	Li et al. dataset	10	99.14%
Chen et al. [30]	Semantic Segmentation and CNN	Grape PlantVillage	4	93.75%
Mishra et al. [31]	CNN	PlantVillage subset + custom images	3	88.46%
Gajjar et al. [32]	SSD combined with CNN	PlantVillage subset	20	96.88%

Fonte: [4]

Os resultados obtidos para a detecção de doenças em folhas de citrus, expostos na Tabela 1, mostram que a rede VGG-16 foi a que obteve maior acurácia, em comparação com os outros 5 algoritmos (89.5%) e, apesar disso, considerando o estado da arte, apresentado na Tabela 2, essa arquitetura não se mostra presente nas avaliações de detecção de doenças em plantas. Vê-se, ainda, na Tabela 2, o uso repetido de bases já consolidadas para a classificação de imagens, como a *PlantVillage*, por exemplo, que foi usada em oito dos dezesseis estudos listados. Em 2020 uma nova base para classificação de imagens foi publicada [12], chamada *PlantDoc* e composta por mais de 2500 imagens e contando com cerca de 300 horas de esforço humano para a rotulagem.

III. MATERIAIS E MÉTODOS

O presente trabalho propõe-se a utilizar 2 algoritmos de Deep Learning, sendo eles um composto pela rede *VGG16-Sequential*, com pesos pré-treinados na base *ImageNet*, e a *CNN-Sequential*, para a detecção de doenças em folhas de plantas presentes na base *PlantDoc* [12]. Tem-se, por objetivo, avaliar o desempenho do melhor modelo de DL exposto na Tabela 1 em comparação com um algoritmo composto por uma *CNN-Sequential* de execução rápida - i.e. que seja capaz de fazer o treinamento de toda a base em menos de 10 minutos - e testar a base *PlantDoc* [12] como uma possível nova base a ser consolidada como referência para modelos de classificação de doenças em folhas.

A. *Base de dados*

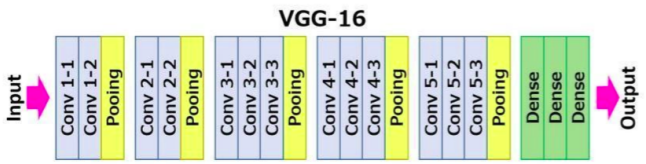
Considerou-se uma base de dados pública, denominada *PlantDoc* [12], que consiste de um conjunto de dados para detecção visual de doenças em plantas, contendo 2.598 imagens no total, referentes a 13 espécies de plantas e 17 classes de doenças, envolvendo aproximadamente 300 horas humanas de esforço em anotar imagens coletadas da internet. A Figura 1 mostra exemplos de imagens da base de dados. Pode-se notar que a base é diversa e conta com uma complexidade pois existem imagens de situações reais e imagens computacionais, inclusive com marcas d'água e escritas. Outro ponto importante a mencionar é que todas as marcações de uma mesma imagem recebem o mesmo rótulo.

Fig. 1. Exemplo de figuras do *dataset* PlantDoc com as regiões anotadas em destaque.

B. *Redes Neurais*

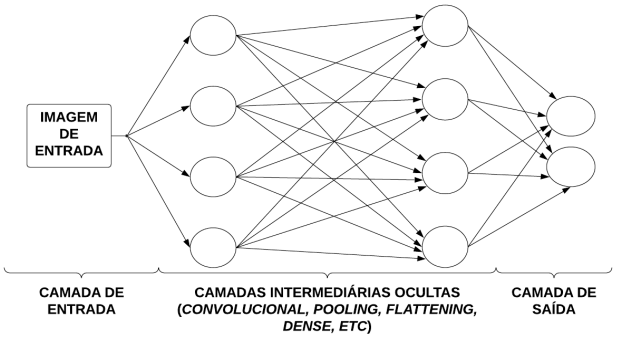
Foram escolhidas as redes *VGG16-Sequential* e a *CNN-Sequential.* A rede *VGG16* foi proposta em 2014 por Simonyan & Zisserman (2014) e é composta por mais de 21 camadas, sendo 16 de parâmetros que podem ser aprendidos, 13 convolucionais e 5 Max Pooling. O tamanho padrão de entrada da rede é (224, 224) com 3 canais RGB. A primeira camada convolucional tem 64 filtros, a segunda tem 128, a terceira, 256 e a quarta e a quinta tem 512 filtros. As três camadas *fully connected* são arquitetadas de forma que as duas primeiras tenham 4096 canais cada e a última tenha 1000 canais - cada um para uma classe. A última camada, por fim, é uma *soft-max*. A Figura abaixo representa a arquitetura de uma VGG16.

Fig. 2. Arquitetura geral de uma rede VGG-16



A CNN-sequential foi inicialmente proposta por Yan Lecun (1998) e é a base de redes conhecidas da literatura, como a AlexNet e a GoogleNet, com 8 camadas e sessenta milhões de parâmetros e vinte e duas camadas e quatro milhões de parâmetros, respectivamente. O principal objetivo dessa rede é filtrar linhas, curvas e bordas, tornando a imagem mais complexa a cada filtragem feita por uma camada. As dimensões de entrada foram (64, 64) com 3 canais RGB. A rede sequential, importada do Keras, API de alto desempenho para redes neurais, conta com duas camadas de convoluções e duas camadas de rede *fully-connected*. com otimizador Stochastic Gradient Descent (SGD) com Learning rate de 0.01.

Fig. 3. Arquitetura geral de uma rede CNN-Sequential



*P*ara o processamento das redes foi usado um *laptop* Dell Latitude 5400 com 32Gb de memória, processador Intel® Core™ i5-8265U CPU @ 1.60GHz × 8 e placa de vídeo Mesa Intel® UHD Graphics 620 (WHL GT2), com sistema operacional Ubuntu 20.04.5 LTS.

IV. Experimentos e Resultados

A. *Limpeza dos dados*

O conjunto de dados possui dois arquivos com os rótulos de cada folha presente nas imagens, sendo um para arquivos de treino e outro para teste. Os arquivos iniciais possuem 8.469 imagens para treino e 452 imagens para testes.

Como a base já é separada para treinamento de modelos e rotulada, apenas duas etapas de limpeza foram aplicadas. A primeira, foi a remoção de elementos cujos nomes dos arquivos eram inválidos. Derivado dessa etapa, 11 imagens (referente a 3 arquivos) foram removidas da base de treino e nenhuma foi removida da base de teste. A segunda foi a remoção de imagens cujas dimensões dos rótulos fossem inválidas, ou seja, imagens nas quais os valores mínimos ou máximos para as coordenadas x e y que designam o recorte rotulado fossem impraticáveis nas dimensões da imagem. Seis regras foram consideradas para que as dimensões fossem consideradas inválidas:

I) xmin >= xmax

II) ymin >= ymax

III) xmin ou xmax <= 0

IV) ymin ou ymax <= 0

V) xmin >= largura ou xmax > largura

VI) ymin >= altura ou ymax > altura.

Após essa validação, 25 imagens foram removidas da base de treino e 3 foram removidas da base de teste.

Por fim, foram consideradas para o treinamento 8.433 imagens, de 29 classes e 13 tipos de doenças. Os modelos foram treinados em duas fases, sendo elas: o treinamento considerando apenas classificação de folhas saudáveis *versus* folhas com alguma doença e o treinamento considerando as 29 classes individualmente. Cada fase teve sua própria preparação e os detalhes estão descritos abaixo.

B. *Separação e pré-processamento dos dados*

Dado que a base já é dividida em treino e teste, foi apenas necessária a divisão da base de treino em treino e validação. Uma das características da base, como pode-se observar na Figura 1 é a de que um mesmo arquivo pode conter múltiplas imagens e, para evitar que imagens provenientes da mesma figura ficassem divididas entre treino e validação (e possivelmente gerassem um viés na validação), a divisão foi feita entre os arquivos e não diretamente entre as imagens. Para a primeira fase dos treinamentos - a saber, considerando apenas duas classes -, a divisão dos arquivos foi feita considerando a estratificação pelo estado de saúde (doente *versus* saudável) e a proporção da base original foi mantida tanto para o treino quanto para a validação: aproximadamente 52% das bases eram possuintes de alguma doença, enquanto 48% eram saudáveis. Exatamente o mesmo procedimento foi realizado para a segunda fase de treino, com a diferença de que a base, nesse caso, foi estratificada considerando a classe da doença, sendo que a classe mais frequente representava aproximadamente 9.5% das imagens e a menos frequente, 0.03% - um desbalanceamento que será discutido posteriormente.

Com as bases já devidamente divididas, as imagens foram redimensionadas e tiveram suas cores convertidas. Vale mencionar que por imagens adota-se o conceito da sub-parte de cada figura correspondente à folha rotulada. Para o modelo VGG-16 as imagens foram redimensionadas para os tamanhos (224, 224) e para o modelo CNN-Sequential elas foram redimensionadas para (64, 64). Para ambos, as cores foram convertidas de BGR (blue, green, red - azul, verde, vermelho) para RGB (vermelho, verde, azul). Nas Figuras 4 e 5 é possível notar que a rotulação das imagens conta com imagens cortadas, sobrepostas e que, para uma mesma doença, existem imagens da frente e do verso da folha, o que gera uma mudança nas características de imagem e cor.

Fig. 4. Exemplo contendo 9 imagens (quadrados destacados em vermelho) referentes à 3 figuras da base de treino antes do processamento

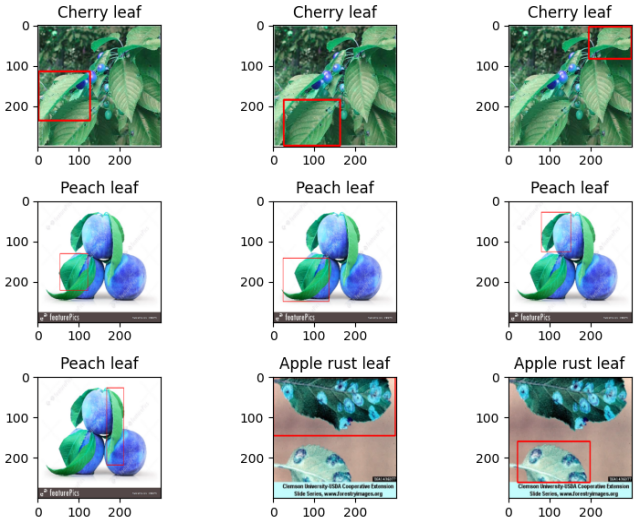
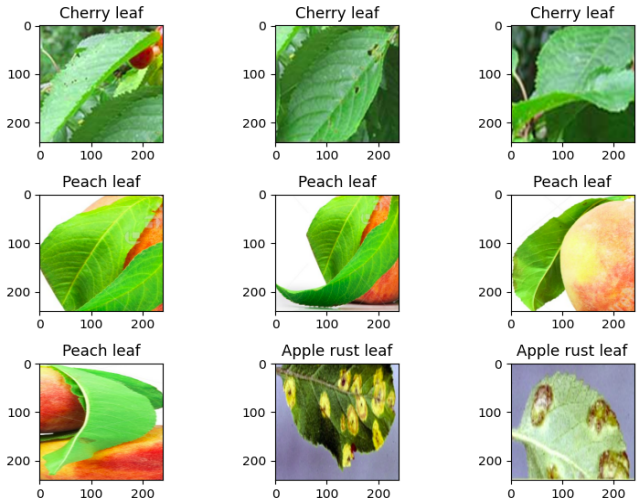


Fig. 5. Exemplo contendo as 9 imagens da figura acima após o processamento (recoloração e redimensionamento para (224,224))



C. *Redes Neurais*

Primeiramente, optamos por usar uma *CNN-Sequential* sem o uso de pesos pré treinados para classificação das imagens entre doentes *versus* saudáveis, ou seja, para uma classificação binária. A rede foi construída usando referências da literatura e o modelo conta com três camadas de aumento de dados (*data augmentation*) , sendo um de flip horizontal, um de rotação (fator=0.1) e um de translação (0.1 largura e 0.1 comprimento), sequenciadas por duas camadas de convoluções e duas camadas de rede *fully-connected*. com otimizador Stochastic Gradient Descent (SGD) com Learning rate de 0.01. As camadas convolucionais foram configuradas com *kernels* de tamanho (4, 4) e 32 filtros, com função de ativação *relu*. A primeira camada *fully-connected* teve função de ativação *relu* e 256 unidades, já a segunda camada teve função de ativação *softmax* e duas unidades - a saber, quantidade de classes, resultando em um modelo com aproximadamente um milhão e quatrocentos mil parâmetros, todos treináveis.

A outra rede escolhida foi uma VGG16 como base do modelo, usando pesos pré treinados com a base "imagenet", sequenciada por mais três camadas *fully-connected*, as duas primeiras com função de ativação *relu* e 128 e 64 unidades, respectivamente, e a última com função de ativação *softmax* e a quantidade de unidades equivalente ao número de classes. O modelo utilizado, composto por esta rede, possui quase dezoito milhões de parâmetros, sendo mais de 3 milhões treináveis (os outros usam os pesos pré treinados- processo conhecido como *transfer-Learning*, já que usa "conhecimentos" adquiridos em outras treinamentos de outras bases como ponto de partida).

O modelo composto pela *CNN-Sequencial* demorou pouco mais de 8 minutos para treinar e teve acurácia de 0.79 no treino e 0.73 na validação. A acurácia da predição na base de testes foi de 0.82, o que mostrou que o modelo não foi treinado com viés, pois foi capaz de ter acurácia similar no treinamento e no teste, com uma base desconhecida. A Figura 6 mostra a matriz de confusão para a classificação das folhas e a Tabela 3 mostra as métricas para avaliação do modelo com a rede *CNN-Sequencial* - as Equações 1, 2 e 3 descrevem as métricas consideradas.

Fig. 6. Matriz de confusão para a classificação binária das folhas usando *CNN-Sequencial*

		Classe Predita	
		Doente	Saudavel
Classe Real	Doente	0.87	0.13
	Saudável	0.23	0.77

TABELA III MÉTRICAS PARA AVALIAÇÃO DO MODELO DE CLASSIFICAÇÃO BINÁRIA DAS FOLHAS USANDO *CNN-SEQUENCIAL*

Classificação	***Precisão***	***Recall***	***f1-score***	***Quantidade de Imagens***
Doente	0.71	0.87	0.78	180
Saudável	0.90	0.77	0.83	269
Acurácia			0.81	449

$$Precisão = \frac{Verdadeiros\ Positivos}{Verdadeiros\ Positivos + Falsos\ Positivos} \quad (1)$$

$$Recall = \frac{Verdadeiros\ Positivos}{Verdadeiros\ Positivos + Falsos\ Negativos} \quad (2)$$

$$F1score = 2 \times \frac{Precisão * Recall}{Precisão + Recall} \quad (3)$$

Podemos notar que, apesar da acurácia de 0.81, a precisão do modelo composto pela *CNN-Sequencial* é de apenas 0.71, ou seja, o modelo classifica como positivo mais dados do que realmente são, ou seja, de cada 100 imagens saudáveis, 23 o modelo classifica como doente - falsos positivos. Considerando o objetivo - identificação de doenças - observa-se pela métrica do *recall* que 87% das imagens com alguma doença são identificadas como doentes. O *f1-score*, por tratar-se de uma média harmônica entre os dois indicadores, fornece uma visão geral do modelo e mostra que o modelo acerta cerca de 78% das folhas com doenças. Um ponto positivo desse modelo foi o

tempo de execução: o modelo foi treinado em menos de 10 minutos para aprender todos os parâmetros utilizados, uma vez que não possui nenhum peso pré treinado e todos os parâmetros são treináveis. Já o modelo composto pela *VGG16-Sequencial* demorou cerca de 262 minutos para treinar, ou seja, quase 4.5 horas, considerando as capacidades de hardware supracitadas. Para esse modelo, a acurácia de treino foi de 0.94 e de validação 0.83. Considerando a base de teste, a acurácia foi de 0.91. A Figura 7 mostra a matriz de confusão para a classificação das folhas e a Tabela 4 mostra as métricas para avaliação do modelo com a rede *VGG16-Sequencial*.

Fig. 7. Matriz de confusão para a classificação binária das folhas usando *VGG16-Sequencial*

		Classe Predita	
		Doente	Saudavel
Classe Real	Doente	0.89	0.11
	Saudável	0.082	0.92

TABELA IV MÉTRICAS PARA AVALIAÇÃO DO MODELO DE CLASSIFICAÇÃO BINÁRIA DAS FOLHAS USANDO *VGG16-SEQUENCIAL*

Classificação	***Precisão***	***Recall***	***f1-score***	***Quantidade de Imagens***
Doente	0.88	0.89	0.89	180
Saudável	0.93	0.92	0.92	269
Acurácia			0.91	449

O modelo com uso de *transfer-Learning* se mostrou mais acurado, mais preciso e mais eficaz do que o modelo anterior. Nesse caso, a predição conta com um precisão de 0.88 para as folhas com doenças e *recall* de 0.89, ou seja, 89% das imagens com doenças são classificadas corretamente. A Tabela abaixo facilita a comparação entre os modelos.

TABELA V RESULTADOS DOS MODELOS PARA DETECÇÃO DE DOENÇAS EM FOLHAS DE PLANTAS

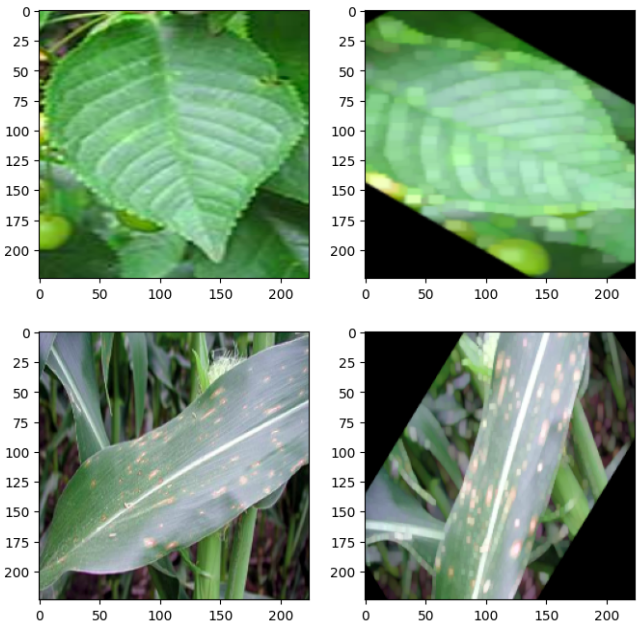
Metricas	***CNN-Sequential***	***VGG16-Sequential***
Pesos pré-treinados	-	"ImageNet"
Parâmetros treináveis	1,403,202	3,219,778
Parâmetros não treináveis	-	14,714,688
Tempo Treinamento (minutos)	8	262
Precisão com Doença	0.71	0.88
Precisão Saudável	0.90	0.93
Recall com Doença	0.87	0.89
Recall Saudável	0.77	0.92
f1-score com Doença	0.78	0.89
f1-score Saudável	0.83	0.92
Acurácia	0.81	0.91

Para a classificação das imagens considerando as vinte e nove classes presentes considerou-se, fez-se necessária a aplicação de mais uma etapa de processamento dos dados, uma vez que as classes são desbalanceadas. A figura abaixo mostra a dimensão do desbalanceamento das classes, considerando a base utilizada para treino e validação. A classe menos frequente possui apenas duas imagens, enquanto a classe mais frequente tem 827. Foram adotadas, portando, as seguintes estratégias:

- As duas classes com menos imagens foram removidas, por não haver imagens suficientes para o treinamento do modelo
- Foi aplicado um processo de balanceamento dos dados, para que a quantidade de imagens por classe fosse mais equilibrada. Esse processo foi guiado pelo terceiro quartil (p75) da seguinte forma: classes com mais imagens do que a quantidade do p75 foram submetidas a *under-sample* e as que tinham menos imagens do que o p75 foram submetidas ao processo de *over-sample.* O *under-sample* foi feito excluindo imagens provenientes da mesma Figura até que a quantidade do p75 fosse atingida. O *over-sample* foi feito em cada classe considerando o mínimo valor entre a quantidade faltante de dados até o p75 e a quantidade de imagens na classe, sendo possível, no máximo, dobrar a quantidade original dos dados de cada classe após essa etapa. O aumento dos dados foi feito através da rotação em 45º e dilatação - transformação morfológica considerando um kernel 5x5 preenchido por 1.
- Como a quantidade de imagens por classe ainda não era igual, o modelo foi treinado considerando pesos diferentes para a penalização da função de perda para cada classe (usando *class_weight*). Vale citar que como o otimizador utilizado é o Adam, não há perigo de falha derivada dos pesos, pois o tamanho de etapa deste otimizador não depende da magnitude do gradiente.

A Figura 8 mostra dois exemplos de folhas utilizadas após o processo de *over-sample*, sendo que na esquerda são as imagens com características originais e na coluna da direita as imagens criadas através da rotação em 45º e dilatação das imagens originais.

Fig. 8. Exemplos de imagens originais (esquerda) e modificadas para o aumento dos dados (direita)



Já as Figuras 9 e 10 permitem ver a quantidade de imagens por classe antes e após o processo de balanceamento de classes, respectivamente. A base utilizada para treino e validação do modelo, após o processamento de balanceamento, tinha um total de 9518 imagens.

Fig. 9. Quantidade de imagens na base usada para treino e validação por classe antes do processo de balanceamento de classes

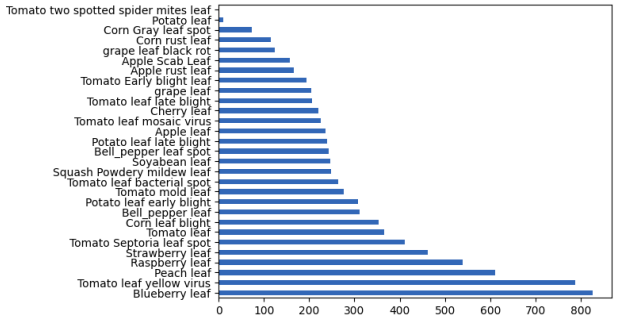
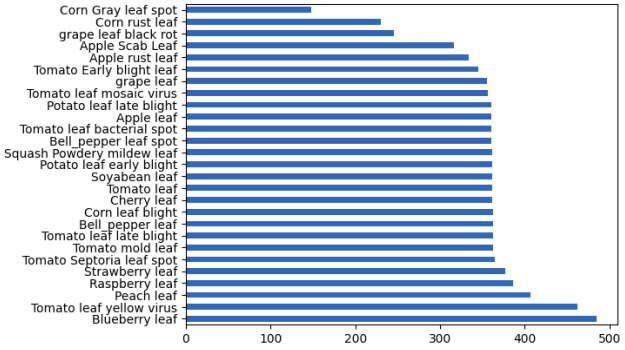


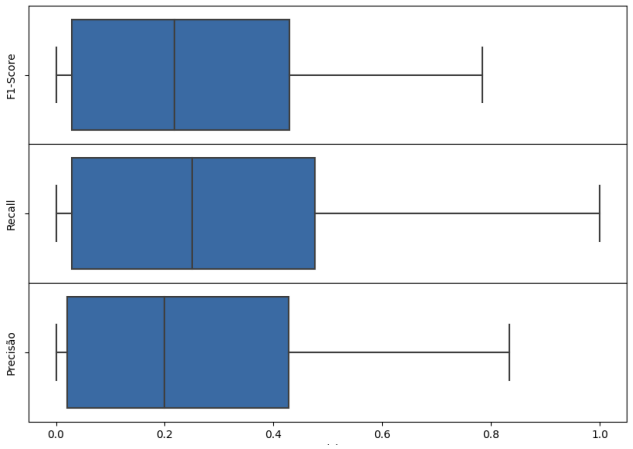
Fig. 10. Quantidade de imagens na base usada para treino e validação por classe após o processo de balanceamento de classes



As arquiteturas de redes usadas foram exatamente as mesmas da classificação binária, com a única diferença das unidades da última camada densa da sequencial ser alterada para 27 - quantidade de classes a serem preditas.

Após essas etapas, obteve-se o treinamento do modelo usando CNN-Sequential, que obteve acurácia de 0.21 no conjunto de teste e o modelo usando *VGG16-Sequential*, que precisou de 435 minutos para ser treinado, ou seja, mais de 7 horas considerando o hardware supracitado, e acurácia total avaliada no conjunto de teste, composto por 449 imagens, de 0.33. Mais informações serão analisadas a seguir a respeito do último modelo. A Figura 11 mostra que para mais de 75% das classes (21 classes) todas as métricas - F1-Score, Recall e Precisão - são inferiores a 0.5, sendo que para metade das classes - 14 classes - as métricas são inferiores a 0.30.

Fig. 11. Distribuição das métricas do modelo VGG166-Sequential l para o diagnóstico de tipos de doenças em folhas de plantas



A classe com maior acurácia foi a strawberry leaf, que obteve F1-score de 0.78, porém com mais falsos negativos do que falsos positivos, ou seja, mais folhas dessa classe foram lassificadas como de outras classes do que folhas de outras classes foram classificadas como desta classe.A Tabela 6 descreve as métricas separadamente por classe.

TABELA VI RESULTADOS DO MODELO USANDO REDE *VGG16-SEQUENTIAL* PARA O DIAGNÓSTICO DE TIPOS DE DOENÇAS EM FOLHAS DE PLANTAS

Classe	***Precisão***	***Recall***	***f1 score***	***nº imagens***
Apple Scab Leaf	0.18	0.15	0.17	13
Apple Leaf	0.15	0.30	0.20	10
Apple rust leaf	0.00	0.00	0.00	10
Bell_pepper leaf	0.04	0.10	0.06	15
Bell_pepper leaf spot	0.00	0.00	0.00	22
Blueberry leaf	0.26	0.59	0.36	19
Cherry leaf	0.46	0.32	0.37	4
Corn Gray leaf spot	0.36	1.00	0.53	12
Corn leaf blight	0.43	0.25	0.32	10
Corn rust leaf	0.83	0.50	0.62	10
Peach leaf	0.35	0.70	0.47	17
Potato leaf early blight	0.20	0.06	0.09	10
Potato leaf late blight	0.00	0.00	0.00	10
Raspberry leaf	0.00	0.00	0.00	17
Soyabean leaf	0.00	0.00	0.00	20
Squash Powdery mildew leaf	0.33	0.17	0.22	6
Strawberry leaf	0.66	0.97	0.78	30
Tomato Early blight leaf	0.00	0.00	0.00	18
Tomato Septoria leaf spot	0.45	0.42	0.43	24

Classe	*Precisão*	*Recall*	*f1 score*	*nº imagens*
Tomato leaf	0.5	0.19	0.27	27
Tomato leaf bacterial spot	0.09	0.29	0.13	14
Tomato leaf late blight	0.16	0.36	0.22	14
Tomato leaf mosaic virus	0.37	0.50	0.42	36
Tomato leaf yellow virus	0.63	0.45	0.53	42
Tomato mold leaf	0.00	0.00	0.00	16
Grape leaf	0.43	0.60	0.50	15
Grape leaf black rot	0.09	0.12	0.11	8

O resultado do modelo pode ter como fatores determinantes a falta de padrão de metodologia na aquisição das fotos por classe do conjunto de treinamento, ou seja, algumas classes contém majoritariamente imagens obtidas de fotos em ambiente controlado, enquanto outras classes possuem majoritariamente imagens obtidas em campo, como pode-se observar nas Figuras 4 e 5; além disso, viu-se que os erros acontecem mais entre classes semelhantes, como os diferentes tipos de doenças das folhas de tomate e entre a classificação de tipos de folhas saudáveis.

V. Conclusão e Trabalhos Futuros

Considerando o objetivo do trabalho - detectar doenças em folhas de plantas usando *Deep Learning* - as redes utilizadas, *CNN-Sequential* e *VGG16-Sequencial* mostraram-se com acurácia de 0.81 e 0.91, respectivamente, considerando as arquiteturas descritas. Como a rede *VGG16* possui mais de 14 milhões de parâmetros, fez--se uso de *transfer-Learning* utilizando pesos treinados com a base "*imagenet"*, com o acréscimo das camadas sequenciais e o uso dessa técnica mostrou-se eficaz para melhorar o desempenho do modelo, o que era esperado uma vez que esses pesos iniciais são bons no processamento de bordas e linhas, o que é útil na detecção das doenças. A tentativa de usar os mesmos modelos para o diagnóstico dos tipos de doenças obteve acurácias de 0.21 para o CNN-Sequential e 0.33 para o *VGG16-Sequential*, considerando o conjunto de testes, sendo que 7 das 27 classes não obtiveram nenhum acerto e a classe com melhores métricas foi a Strawberry leaf, uma classe de folhas saudáveis, que obteve outras classes folhas saudáveis classificadas como sendo parte dessa classe (falsos positivos). Essa parte da análise foi útil, porém, para identificar-se a importância do uso de técnicas de balanceamento de classe, uma vez que a execução do modelo sem o uso dessas técnicas causa o *overfitting* dos dados. A base *PlantDoc* [12] mostrou-se efetiva para a utilização em classificações de doenças em folhas de plantas, mas por ter imagens reais e imagens computadorizadas misturadas, a identificação de algumas classes é prejudicada, pois, em situações reais, a mesma imagem consta com folhas portadoras de doenças e folhas saudáveis, já que a rotulagem foi feita considerando a imagem como um todo. Os resultados poderiam ser mais acurados após um novo esforço de rotulagem baseado em cada folha especificamente, com seus rótulos individuais, através da segmentação por instâncias.

Para futuros estudos nesta área, propõe-se o uso de outras técnicas de *data augmentation,* uso de pesos treinados em outras bases de dados e de outras arquiteturas de redes conhecidas, como a *VGG19*, por exemplo. Sugere-se também, para a melhoria da classificação multiclasse, o uso de modelos de detecção de objetos para que as marcações dos rótulos das folhas sejam aperfeiçoados sem a necessidade de novos esforços humanos e o uso de segmentação semântica, para identificação das particularidades de cada doença, uma vez que se viu uma concentração de erro entre folhas de classes semelhantes (diferentes doenças nas folhas de tomate, por exemplo).

Referências

[1] BECKMAN, J., & COUNTRYMAN, A. M. (2021). The Importance of Agriculture in the Economy: Impacts from COVID‐19. American journal of agricultural economics, 103(5), 1595-1611. Disponível em: https://onlinelibrary.wiley.com/doi/full/10.1111/ajae.12212. Acesso em 10 de Janeiro de 2023.

[2] CEPEA - Centro de Estudos Avançados em Economia Aplicada. Pib do agronegócio cresceu abaixo das projeções. CEPEA-USP/CNA, 2022. Disponível em: https://www.cepea.esalq.usp.br/upload/kceditor/files/Cepea_CNA_PIB_JAn_Dez_2021_Mar%C3%A7o2022.pdf. Acesso em: 10 de Janeiro 2022.

[3] CHITRADEVI, B.; SRIMATHI, P. An overview on image processing techniques. International Journal of Innovative Research in Computer and Communication Engineering, v. 2, n. 11, p. 6466-6472, 2014. Disponível em https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.1086.6403&rep=rep1&type=pdf. Acesso em 04 de junho de 2022.

[4] FALASCHETTI, Laura et al. A CNN-based image detector for plant leaf diseases classification. Hardware Xv. 12, p. e00363, 2022. Disponível em https://www.sciencedirect.com/science/article/pii/S2468067222001080. Acesso em 14 de Janeiro de 2023.

[5] FAO - Food And Agriculture Organization Of The United Nations. Statistics Division. World Food and Agriculture Statistical Yearbook 2021. Food and Agriculture Organization of the United Nations, 2021. Disponível em https://www.fao.org/documents/card/en/c/cb4477en/. Acesso em 4 de junho de 2022.

[6] LECUN, Yann et al. Gradient-based Learning applied to document recognition. Proceedings of the IEEE, v. 86, n. 11, p. 2278-2324, 1998. Disponível em http://yann.lecun.com/exdb/publis/pdf/lecun-98.pdf. Acesso em 15 de Janeiro de 2023.

[7] LIAKOS, K. G et al. 2018. Machine Learning in agriculture: A review. Sensors 18, no. 8 (2018): 2674. Disponível em: https://www.mdpi.com/1424-8220/18/8/2674/htm. Acesso em 25 de Novembro de 2022.

[8] SALVI, M. et al. The impact of pre-and post-image processing techniques on Deep Learning frameworks: A comprehensive review for digital pathology image analysis. Computers in Biology and Medicine, v. 128, p. 104129, 2021. Disponível em https://www.sciencedirect.com/science/article/pii/S0010482520304601. Acesso em 04 de junho de 2022.

[9] SAVARY, S. et al. Crop losses due to diseases and their implications for global food production losses and food security. Food security vol. 4, 2012, p. 519-537. Disponível em: https://link.springer.com/article/10.1007/s12571-0. Acesso em 25 de Novembro de 2022.

[10] SHORTEN, C.; KHOSHGOFTAAR, T. M. A survey on image data augmentation for Deep Learning. Journal of big data, v. 6, n. 1, p. 1-48, 2019. Disponível em https://link.springer.com/article/10.1186/s40537-019-0197-0?code=a6ae644c-3bfc-43d9-b292-82d77d5890d5. Acesso em 04 de junho de 2022.

[11] SIMONYAN, Karen; ZISSERMAN, Andrew. Very Deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014. Disponível em https://arxiv.org/abs/1409.1556. Acesso em 14 de Janeiro de 2023.

[12] SINGH, D. et al. Plantdoc: a dataset for visual plant disease detection. In: Proceedings of the 7th ACM IKDD CoDS and 25th COMAD. 2020. p. 249-253. Disponível em https://arxiv.org/pdf/1911.10317.pdf. Acesso em 04 de junho de 2022.

[13] SUJATHA, R. et al. 2021. Performance of Deep Learning vs machine Learning in plant leaf disease detection. Microprocessors and Microsystems, v. 80, p. 103615, 2021. Disponível em https://www.sciencedirect.com/science/article/abs/pii/S0141933120307626. Acesso em 04 de junho de 2022.

[14] WEISS, K.; KHOSHGOFTAAR, T. M.; WANG, D.. A survey of transfer Learning. Journal of Big data, v. 3, n. 1, p. 1-40, 2016. Disponível em https://journalofbigdata.springeropen.com/articles/10.1186/s40537-016-0043-6.Acesso em 10 de Janeiro de 2023.

[15] YANG, X., & GUO, T., 2017. Machine Learning in plant disease research. Biomedical Research Journal, vol. 34, 2019. Disponível em: https://biomedicaljour.com/pdfs/volume-34/8.pdf. Acesso em 10 de Janeiro de 2023.

[16] ZHAI, Z. et al. 2020. Decision support systems for agriculture 4.0: Survey and challenges. Computers and Electronics in Agriculture, 170, 105256. Disponível em https://www.sciencedirect.com/science/article/pii/S0168169919316497#b0005. Acesso em 04 de junho de 2022.

[17] MOHANTY, D.P at al. 2016. Using deep learning for image-based plant disease detection. Front. Plant Sci., 7 (2016), p. 1419. Disponível em https://www.frontiersin.org/articles/10.3389/fpls.2016.01419/full. Acesso em 25 de Janeiro de 2023.

[18] RAMCHARAN, K. et al. 2017. Deep Learning for Image-Based Cassava Disease Detectio. Front. Plant Sci., 8 (2017), p. 1852. Disponível em https://doi.org/10.3389/fpls.2017.01852. Acesso em 25 de Janeiro de 2023.

[19] FUENTES, S. et al. 2017. A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition. Sensors, 17 (9) (2017), p. 2022. Disponível em https://www.mdpi.com/1424-8220/17/9/2022. Acesso em 25 de Janeiro de 2023.

[20] PAWARA, E. P. et al. 2017. Comparing local descriptors and bags of visual words to deep convolutional neural networks for plant recognition. International Conference on Pattern Recognition Applications and Methods, SciTePress, 2 (2017), pp. 479-486. Disponível em https://doi.org/10.5220/0006196204790486. Acesso em 25 de Janeiro de 2023.

[21] FERENTINOS, K. P. et al. 2018. Deep learning models for plant disease detection and diagnosis. Comput. Electron. Agricult., 145 (2018), pp. 311-318. Disponível em https://doi.org/10.1016/j.compag.2018.01.009. Acesso em 25 de Janeiro de 2023.

[22] RAMCHARAN, P. A. et al. 2019. A mobile-based deep learning model for cassava disease diagnosis. Front. Plant Sci., 10 (2019), p. 272. Disponível em https://doi.org/10.3389/fpls.2019.00272. Acesso em 25 de Janeiro de 2023.

[23] GEETHARAMANI, G. A. PANDIAN J., 2019, Identification of plant leaf diseases using a nine-layer deep convolutional neural network. Comput. Electr. Eng., 76 (2019) 323-338. doi: 10.1016/j.compeleceng.2019.04.011.

[24] CHEN, J. et al, 2020. Using deep transfer learning for image-based plant disease identification. Comput. Electron. Agricul., 173 (2020). Disponível em https://doi.org/10.1016/j.compag.2020.105393. Acesso em 25 de Janeiro de 2023.

[25] CHEN, J. et al, 2021. Attention embedded lightweight network for maize disease recognition.Plant. Pathol., 70 (3) (2021), pp. 630-642. Disponível em https://doi.org/10.1111/ppa.13322. Acesso em 25 de Janeiro de 2023.

[26] CHEN, J. et al, 2021. Identifying crop diseases using attention embedded MobileNet-V2 model. Appl. Soft Comput., 113. Disponível em https://doi.org/10.1016/j.asoc.2021.107901. Acesso em 25 de Janeiro de 2023.

[27] CHEN, J. et al, 2021. Automatic identification of commodity label images using lightweight attention network. Neural Comput. Appl., 33 (2021). Disponível emhttps://doi.org/10.1007/s00521-021-06081-9. Acesso em 25 de Janeiro de 2023.

[28] Li, Y. et al, 2020. Few-shot cotton pest recognition and terminal realization. Comput. Electron. Agricult., 169 (2020), Article 105240. Disponível em https://doi.org/10.1016/j.compag.2020.105240. Acesso em 25 de Janeiro de 2023.

[29] CHEN, J. W. et al, 2021. Crop pest recognition using attention-embedded lightweight network under field conditions. Appl. Entomol. Zool., 56 (2021), pp. 427-442. Disponível em https://doi.org/10.1007/s13355-021-00732-y. Acesso em 25 de Janeiro de 2023.

[30] CHEN, J. et al, 2021. A cognitive vision method for the detection of plant disease images. Mach. Vis. Appl., 32 (2021), p. 31. Disponível em https://doi.org/10.1007/s00138-020-01150-w. Acesso em 25 de Janeiro de 2023.

[31] MISHRA, S., R. SACHAN, D. RAJPAL, 2020. Deep Convolutional Neural Network based Detection System for Real-time Corn Plant Disease Recognition. Procedia Comput. Sci., 167 (2020), pp. 2003-2010. Disponível em https://www.sciencedirect.com/science/article/pii/S2468067222001080#b0195. Acesso em 25 de Janeiro de 2023.

[32] GAJJAR, R. N. et al, 2021. Real-time detection and identification of plant leaf diseases using convolutional neural networks on an embedded platform. Visual Comput. (2021). Disponível em https://doi.org/10.1007/s00371-021-02164-9. Acesso em 25 de Janeiro de 2023.

COMPARAÇÃO DE DESEMPENHO DE BANCOS DE DADOS SQL E NoSQL

Omar Hajime Fidelis
Pós-Graduação em Ciência de Dados
Instituto Federal de Educação, Ciência e Tecnologia de São Paulo – Câmpus Campinas
Campinas, Brasil
omar_fidelis@yahoo.com

Prof. Dr. Andreiwid Sheffer Corrêa
Pós-Graduação em Ciência de Dados
Instituto Federal de Educação, Ciência e Tecnologia de São Paulo – Câmpus Campinas
Campinas, Brasil
http://orcid.org/0000-0003-2943-0111

***Abstract*— A geração de quantidades enormes de dados requer ferramentas para seu armazenamento e desempenho para tomada de decisão. Este estudo apresenta a análise e o comparativo entre dois bancos de dados. A escolha dos Sistemas de Gerenciamento de Banco de Dados foi motivada pela possibilidade de comparar o desempenho do MySQL, um banco de dados relacional (SQL), com o MongoDB, um banco de dados não-relacional (NoSQL). Ambos possuem distribuição gratuita e, estão classificados entre os cinco melhores bancos de dados do mercado. Este trabalho realiza uma série de testes nos bancos de dados para a coleta do tempo de execução, e assim, compara o desempenho dos dois SGBD para atividades comumente utilizadas.**

Keywords*— *comparação, banco de dados, desempenho, MySQL, MongoDB

I. INTRODUÇÃO

Atualmente, uma grande quantidade de dados está sendo gerada a partir de variadas fontes de dados, como smartphones, computadores, sensores, câmeras, sistemas de posicionamento global, sites de redes sociais, transações comerciais e jogos [5]. Com a ajuda de ferramentas de análise é possível extrair a partir desses dados inúmeras informações e aplicá-las no dia a dia. Contudo, essa não é uma tarefa fácil. Um dos desafios mais importantes para a comunidade de pesquisadores de banco de dados nos últimos anos tem sido o desenvolvimento de tecnologias para gerenciar essa grande quantidade de dados heterogêneos produzidos em uma alta velocidade por aplicativos e pessoas [16].

Os bancos de dados relacionais, também conhecidos como banco de dados SQL ("Structured Query Language", ou "Linguagem de Consulta Estruturada"), são os mais utilizados no mundo, mas por serem desenvolvidos sob uma estrutura de relacionamento bem definida possuem limitações relacionadas ao gerenciamento de grande volume dados heterogêneos [1]. Isso incentivou a comunidade de banco de dados a buscar novas soluções de armazenamento para esses dados. Uma dessas soluções são os Sistemas de Gerência de Banco de Dados (SGBDs) NoSQL ("Not Only SQL") [4]. Ao contrário do que muito se especula, a tecnologia NoSQL não veio com o intuito de abolir as tecnologias relacionais, mas como uma alternativa para suprir suas limitações [4].

Os sistemas NoSQL trabalham com armazenamento distribuído, não utilizando o modelo de dados relacional e nem sempre a linguagem SQL [7]. Estes sistemas podem ser subdivididos de acordo com a técnica de armazenamento, sendo os quatro principais: NoSQL chave-valor, NoSQL modelo Colunar, NoSQL orientado a documento e NoSQL representado por Grafos [17].

Um fator que vem se destacando como ponto decisivo para escolha de um sistema é o desempenho, isto é, o tempo de resposta para atender as necessidades das aplicações, e consequentemente dos usuários, principalmente, considerando o crescimento constante do volume de dados que os SGBDs precisam gerenciar [13]. A escolha de um Sistema Gerenciador de Banco de Dados (SGBD), dentre a grande diversidade disponível no mercado, é uma tarefa delicada, devido a importância e responsabilidade que essa ferramenta representa por gerenciar uma das maiores riquezas de uma organização, que são as informações.

Considerando este cenário, o objetivo deste trabalho é um comparativo de desempenho de leitura e escrita entre os bancos de dados MySQL (SQL) e MongoDB (NoSQL), com base no tempo de execução de operações comuns em banco de dados com diferente volume de dados.

II. BANCO DE DADOS

Um banco de dados é uma coleção organizada de informações - ou dados - estruturadas, normalmente armazenadas eletronicamente em um sistema de computador. Um banco de dados é geralmente controlado por um sistema de gerenciamento de banco de dados (SGBD ou, em inglês, DBMS) [10]. Os dados, o SGBD e os aplicativos associados a eles também são chamados de sistema de banco de dados.

Os bancos de dados mais comuns no mercado representam o armazenamento dos dados em linhas e colunas em uma série de tabelas para permitir que sejam facilmente acessados, gerenciados, modificados, atualizados, controlados e organizados. A maioria dos bancos de dados utiliza a linguagem de consulta estruturada (SQL) ou uma variação para permitir o gerenciamento do SGBD e a manipulação dos dados.

A. Banco de Dados Relacional (SQL)

O modelo de dados relacional foi desenvolvido para padronizar a representação e as consultas aos dados, e assim permitindo o acesso por qualquer aplicativo de forma controlada e esperada. Desde o início, os desenvolvedores entenderam que a o foco do modelo de banco de dados relacional estaria no uso de tabelas, uma maneira intuitiva, eficiente e flexível de armazenar e acessar informações estruturadas [14].

A linguagem de consulta estruturada (SQL) tem sido amplamente utilizada como a linguagem para consultas de

banco de dados. Com base na álgebra relacional, a SQL fornece uma linguagem matemática internamente consistente que facilita a melhoria do desempenho de todas as consultas ao banco de dados [9].

B. *Banco de Dados Não Relacional (NoSQL)*

Os bancos de dados NoSQL (Not only SQL) são banco de dados não relacionais, e são amplamente usados em aplicativos da web em tempo real e big data, porque suas principais vantagens são alta escalabilidade e alta disponibilidade.

Os bancos de dados NoSQL são a escolha preferida dos desenvolvedores, pois eles naturalmente aceitam um paradigma de desenvolvimento ágil, adaptando-se rapidamente aos requisitos em constante mudança [11].

Os bancos de dados NoSQL permitem que os dados sejam armazenados de maneiras mais intuitivas e fáceis de entender, ou mais próximas da maneira como os dados são usados pelos aplicativos - com menos transformações necessárias ao armazenar ou recuperar usando APIs no estilo NoSQL. Além disso, os bancos de dados NoSQL podem aproveitar ao máximo a nuvem para oferecer tempo de inatividade zero [4].

III. Medição de Desempenho

Uma prática comum de mercado é a análise de desempenho utilizando como parâmetro a medição de tempo para a execução de uma atividade, ou na execução de uma quantidade de atividades em um período definido (por exemplo, transações de um banco de dados por segundo).

Abaixo, seguem trabalhos e estudos relacionados, que utilizaram o tempo como medição comparativa de desempenho.

A. *Comparativo de Desempenho entre Bancos de Dados de Código Aberto*

No trabalho "Comparativo de Desempenho entre Bancos de Dados de Código Aberto" [12], foram realizados testes comparativos de desempenho para os bancos de dados PostgreSQL e MySQL, sendo que o desempenho do PostgreSQL foi superior ao MySQL apenas no módulo de carga e estrutura.

B. *A performance comparison of SQL and NoSQL databases*

No trabalho "A performance comparison of SQL and NoSQL databases" [6], apresenta tabelas comparativas do tempo de execução de processos de leitura, escrita, deleção e "fetching" para os bancos de dados MongoDB, RavenDB, CouchDB, Cassandra, Hypertable, Couchbase, e MS SQL Express.

C. *Análise de desempenho de Bancos de Dados*

No trabalho "Análise de desempenho de Bancos de Dados" [3], os resultados foram divididos em 3 grupos de acordo com o volume de registros: 1.000, 10.000 e 100.000. Em cada grupo, foram realizadas as operações de inserção, consulta, alteração e exclusão para os bancos de dados MySQL, Firebird, PostgreSQL e SQL Server 2008, e os resultados apresentados são os tempos médios de execução.

D. *Análise comparativa de desempenho de consultas entre um de banco de dados relacional e um banco de dados não relacional*

No trabalho "Análise comparativa de desempenho de consultas entre um de banco de dados relacional e um banco de dados não relacional" [15], os resultados foram divididos em 6 grupos de acordo com o volume de registros: 1 mil, 10 mil, 100 mil, 1 milhão, 2 milhões e 4 milhões. Em cada grupo, foram realizadas as atividades agregação para os bancos de dados SQL Server e MongoDB, e os resultados apresentados são os tempos médios de execução do comando, com melhor desempenho para o banco de dados não-relacional.

IV. Método

A. *Configuração do Ambiente*

A infraestrutura utilizada na execução dos testes e coleta dos resultados deste trabalho considerou a utilização de virtualização para fornecer um servidor com capacidade dedicada, de forma a minimizar a influência de outros processos.

Abaixo, seguem as características do computador base do experimento.

TABELA I. Configuração de hardware e software

Item	Descrição
Computador	Lenovo ThinkPad T495 (modelo 20NK)
Sistema Operacional	Microsoft Windows 10 Enterprise
Plataforma	64-bit operating system, x64-based processor
Processador	AMD Ryzen 5 PRO 3500U w/ Radeon Vega Mobile Gfx 2.10 GHz (4 Cores, 8 Logical Processors)
Memória	16GB (2 x 8GB - DD4 - 2667 MHz)
Disco	SSD 256GB M.2 2280 PCIe Gen3 x4 NVMe

A configuração do ambiente virtual dedicado seguirá a última versão dos softwares disponíveis no momento dos testes.

TABELA II. Configuração do ambiente virtual

Item	Descrição
Software de virtualização	Oracle VM Virtual Box (versão 6.1.34)
Memória	8GB (reservado para partição virtual)
Disco	64GB (reservado para partição virtual)
Sistema Operacional	CentOS Stream (versão 9 – x86_64)
MySQL	MySQL Community Server 8.0.29 for LINUX (x86, 64-bit)
MongoDB	MongoDB 5.0 Community Edition on RHEL / CentOS

A escolha do ambiente LINUX como sistema operacional para a instalação dos bancos de dados e execução dos testes levou em consideração a estabilidade da plataforma, e principalmente as ferramentas embarcadas que permitem a coleta dos resultados e monitoramento do ambiente durante os testes.

B. Base de Dados

A base de dados que será utilizada para a execução dos testes foi a "T-Drive trajectory data sample" desenvolvida por Yu Zheng. Esta base foi escolhida por estar disponível na internet de forma gratuita, ser amplamente utilizada, e possuir um grande volume de dados para obter melhor representatividade nos testes.

A amostra do conjunto de dados da "T-Drive trajectory" que contém trajetórias de uma semana de 10.357 táxis. O número total de pontos neste conjunto de dados é de cerca de 15 milhões e a distância total das trajetórias chega a 9 milhões de quilômetros [18].

C. Linha de comando shell

Os bancos de dados MySQL e MongoDB possuem linha de comando em UNIX, que permitem a execução de lotes de comandos que interagem com os dados armazenados, não requerendo a interação manual.

O MySQL possui o comando mysqlsh e MongoDB possui o comando mongo, que permitem a execução de um arquivo no padão Javascript, o qual contém a lista de comandos a serem executados no banco de dados para cada um dos testes.

Abaixo, segue formato do comando a ser utilizado no MySQL:

```
> mysqlsh --uri <user>:<password>@<host>/<database>
--file <myjsfile.js>
```

Abaixo, segue formato do comando a ser utilizado no MongoDB:

```
> mongo <host>:<port>/<database> <myjsfile.js>
```

D. Comandos para os testes

A métrica definida para comparativo de desempenho entre os bancos de dados é o tempo de resposta para determinadas atividades comumente realizadas no mercado.

Abaixo, seguem os comandos a serem executados:

TABELA III. COMANDOS POR BANCO DE DADOS

Operação	MySQL	MongoDB
Procura	SELECT <campo> FROM <tabela> WHERE <campo> = <valor>	db.<coleção>.find({ <campo>: <valor> } , { <campo>:1 })
Inserção	INSERT INTO <tabela> (<campos>) VALUES (<valores>)	db.<coleção>.insert({ <campo>: <valor>, <campo>: <valor> })
Remoção	DELETE FROM <tabela> WHERE <campo> = <valor>	db.<coleção>.remove({ <campo>: <valor> })
Atualização	UPDATE <tabela> SET <campo> = <valor> WHERE <campo> = <valor>	db.<coleção>.update({ <campo> : <valor> }, {$set: { <campo>:<valor> } } , {multi : true})

V. DESENVOLVIMENTO

Uma vez o ambiente virtual instalado e configurado, realizou-se a carga inicial dos bancos de dados MySQL e MongoDB, utilizando a base de dados "T-Drive trajectory data sample", e para padronizar os testes, a carga inicial considerou os primeiros 10 milhões de registros.

O teste de desempenho foi realizado em 2 etapas distintas:

A. Preparação

A etapa de preparação consiste basicamente na criação dos arquivos Javascript a serem usados para os testes de desempenho.

Utilizando a shell script do UNIX e o comando "random", foram selecionados de forma aleatória os registros a serem utilizados em cada teste de desempenho.

Para os testes de "procura", "remoção" e "atualização", foram selecionados registros da carga inicial, e para os testes de "inserção", foram selecionados registros da base de dados "T-Drive trajectory data sample" que não foram carregados na carga inicial.

Os arquivos Javascript foram criados com 1 mil, 10 mil e 100 mil de registros, configurado conforme o banco de dados (MySQL ou MongoDB) e a operação executada (procura, inserção, remoção e atualização).

B. Testes

Os testes para a análise de desempenho foram realizados utilizando shell script do UNIX e o comando "time" para medir o tempo de execução de lote de comandos nos bancos de dados, portanto a medição foi restrita a execução do comando, excluindo a preparação dos arquivos Javascript.

Para cada banco de dados (MySQL e MongoDB), operação executada (procura, inserção, remoção e atualização) e volume de registros (1 mil, 10 mil e 100 mil de registros), foram realizadas 100 execuções para obtenção de amostragem significativa para estudos dos resultados.

VI. RESULTADOS

Os tempos obtidos para execução das operações mostraram-se consistentes para os dois bancos de dados, sendo as operações de remoção e atualização que consumiram mais tempo, e a inserção a que consumiu menos tempo.

Apesar do aumento substancial do número de registros, o aumento do tempo das operações não se mostrou proporcional.

Abaixo, seguem as médias dos resultados obtidos nos testes para análise de desempenho. Todos os tempos estão em segundos.

TABELA IV. RESULTADOS (SEGUNDOS)

Operação	Volume Registros	MySQL	MongoDB
Procura	1.000	5,291	4,358
	10.000	11,773	9,823
	100.000	23,939	20,201
Inserção	1.000	2,088	1,404
	10.000	6,881	5,744
	100.000	14,611	12,883
Remoção	1.000	21,188	20,004
	10.000	45,394	42,891
	100.000	97,100	92,705
Atualização	1.000	19,886	18,315
	10.000	42,191	37,94
	100.000	87,773	78,004

Os gráficos dos tempos médios das operações nos dois bancos de dados mostram um padrão semelhante em todos os casos.

FIGURA I. OPERAÇÃO: PROCURA

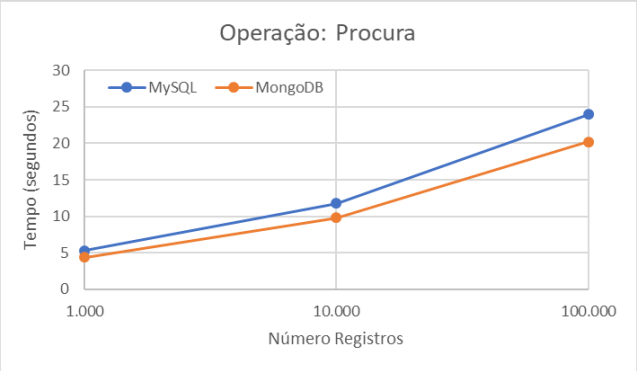


FIGURA II. OPERAÇÃO: INSERÇÃO

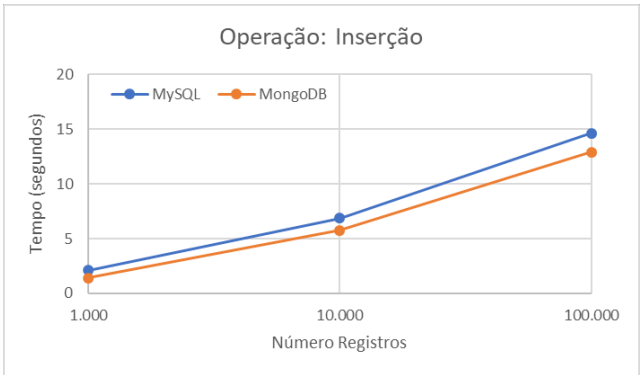


FIGURA III. OPERAÇÃO: REMOÇÃO

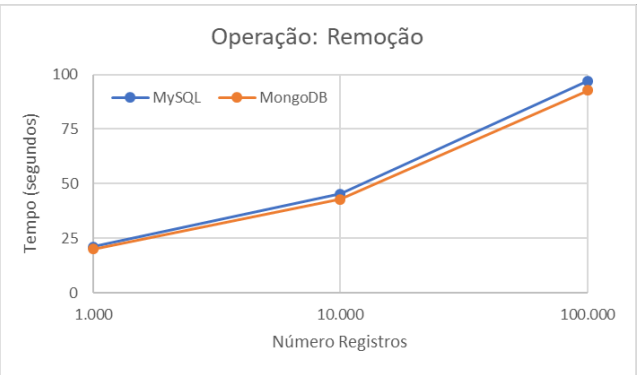
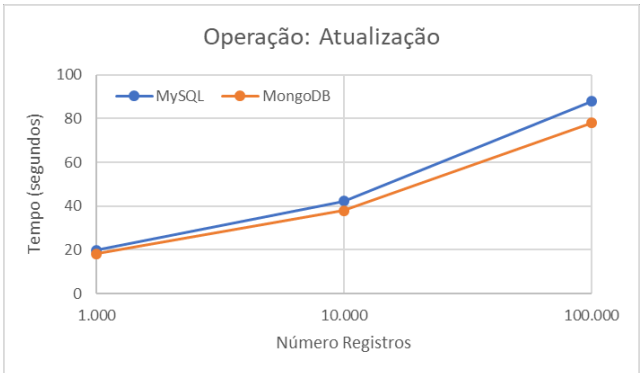


FIGURA IV. OPERAÇÃO: ATUALIZAÇÃO



VII. CONCLUSÃO

A infraestrutura utilizada neste estudo apresentou-se estável, de forma que a variação dos dados obtidos dos testes foi mínima.

O banco de dados MongoDB apresentou menor tempo de execução das operações em relação ao MySQL em todos cenários contemplados neste estudo. Contudo, o desempenho não é o único critério que diferencia os dois bancos de dados.

Abaixo, seguem alguns critérios importantes e diferentes entre os bancos de dados.

TABELA V. COMPARATIVO ENTRE MYSQL E MONGODB [8]

Critérios	MySQL	MongoDB
Tipo de Dado	Dado Estruturado	Dado Estruturado e/ou Não Estruturado
Representação dos dados	Dados em tabelas e linhas	Dados como documentos JSON
Definição de Esquema	Necessário definir tabelas e colunas	Não é necessário esquema
Operações de JOIN	Suporta	Não Suporta
Linguagem de comandos	Structured Query Language(SQL)	MongoDB Query Language (MQL)
Escalabilidade	Escalabilidade limitada	Escalável

Portanto, a escolha do banco de dado não deve considerar apenas um critério, como o desempenho, mas, no caso de empresas, deve-se considerar o conhecimento da equipe técnica, uma vez que a linguagem SQL é mais comum entre os especialistas no mercado.

Outro ponto fundamental é a aplicação conectada ao banco de dados e, portanto, o tipo de dado que será armazenado. No caso de aplicações Web, normalmente os dados não apresentam uma estrutura definida e são estruturados no formato JSON, o que favorece a escolha do banco de dados MongoDB.

Quando os dados possuem um relacionamento bem definido, e podem ser estruturados em tabelas com colunas, o banco de dados MySQL apresenta uma facilidade para a estrutura e armazenamento dos dados.

Para futuro estudos, pode-se realizar análises comparativas de uso de recurso computacional, como memória e disco, além de explorar outras técnicas inerentes de cada banco de dados, como índice, que melhora a consulta dos dados.

REFERÊNCIAS BIBLIOGRÁFICAS

[1] DATE, Chris J. Database in Depth: Relational Theory for Practitioners. Michigan: O'Reilly Media, 2005.

[2] DB-ENGINES Ranking. Disponível em: https://db-engines.com/en/ranking. Acesso em: 02 jun. 2022.

[3] FERREIRA, Erick Rodrigues; TRAD JUNIOR, Sergio M.. Analise de desempenho de Bancos de Dados. Disponível em: https://ri.unipac.br/repositorio/wp-content/uploads/2019/07/Erick-Rodrigues-Ferreira.pdf. Acesso em: 05 jun. 2022.

[4] FOWLER, Martin J.; SADALAGE, Pramodkumar J.. Nosql Distilled: A Brief Guide to the Emerging World of Polyglot Persistence. New Jersey: Addison-Wesley Professional, 2013.

[5] HASHEM, Ibrahim Abaker Targio; CHANG, Victor; ANUAR, Nor Badrul; ADEWOLE, Kayode; YAQOOB, Ibrar; GANI, Abdullah; AHMED, Ejaz; CHIROMA, Haruna. The role of big data in smart city.

International Journal of Information Management. Southampton, p. 748-758. 2016.

[6] LI, Yishan; MANOHARAN, Sathiamoorthy. A performance comparison of SQL and NoSQL databases. Disponível em: https://www.researchgate.net/publication/261079289_A_performance_comparison_of_SQL_and_NoSQL_databases. Acesso em: 05 jun. 2022.

[7] MAYER-SCHONBERGER, Viktor; CUKIER, Kenneth. Big Data: A Revolution That Will Transform How We Live, Work, and Think. 2. ed. Boston: Eamon Dolan/Houghton Mifflin Harcourt, 2014.

[8] MEHER, Easha. MongoDB vs MySQL Performance: 7 Critical Differences. 2021. Disponível em: https://hevodata.com/learn/mongodb-vs-mysql/. Acesso em: 28 jan. 2023.

[9] O QUE É um Banco de Dados Relacional? Disponível em: https://www.oracle.com/br/database/what-is-a-relational-database/. Acesso em: 04 jun. 2022.

[10] O QUE É um Banco de Dados? Disponível em: https://www.oracle.com/br/database/what-is-database/. Acesso em: 04 jun. 2022.

[11] O QUE é um intervalo de confiança? Minitab 19. Disponível em: https://support.minitab.com/pt-br/minitab/19/help-and-how-to/statistics/basic-statistics/supporting-topics/basics/what-is-a-confidence-interval/. Acesso em: 02 jun. 2022.

[12] PIRES, Carlos E. S.; NASCIMENTO, Rilson O.; SALGADO, Ana C.. Comparativo de Desempenho entre Bancos de Dados de Código Aberto. Disponível em: http://www.itautec.com.br/ifileexplorer/arquivo/empresa/documentos/comparativo_%20bancos%20de%20dados_%20sw%20livre.pdf. Acesso em: 05 jun. 2022.

[13] SCALZO, Bert; KLINE, Kevin; FERNANDEZ, Claudia; AULT, Mike; BURLESON, Donald. Database Benchmarking: Practical Methods for Oracle & SQL Server (IT In-Focus series). Kittrell: Rampant Techpress, 2007.

[14] SILBERSCHATZ, Abraham; KORTH, Henry F.; SUDARSHAN, S.. Sistema de Banco de Dados. 7. ed. Rio de Janeiro: Gen Ltc, 2020.

[15] SILVA, Gilmar José da; FERREIRA, Júlio Cesar Oliveira. Análise comparativa de desempenho de consultas entre um de banco de dados relacional e um banco de dados não relacional. Disponível em: https://repositorio.uniube.br/bitstream/123456789/178/1/Gilmar%20Jos%C3%A9%20da%20Silva%20e%20J%C3%BAlio%20Cesar%20Oliveira%20Ferreira.PDF. Acesso em: 05 jun. 2022.

[16] STONEBRAKER, Michel. What Does 'Big Data' Mean? 2012. Disponível em: http://cacm.acm.org/blogs/blog-cacm/155468-what-does-big-data-mean/fulltext. Acesso em: 19 abr. 2022.

[17] TURMASM. Tipos de bancos de dados NoSQL. 2017. Disponível em: https://micreiros.com/tipos-de-bancos-de-dados-nosql/. Acesso em: 19 abr. 2022.

[18] ZHENG, Yu; YUAN, Jing; SUN, Guangzhong; XIE, Xing. T-Drive trajectory data sample. 2011. Disponível em: https://www.microsoft.com/en-us/research/publication/t-drive-trajectory-data-sample/. Acesso em: 02 jun. 2022.

Algoritmos de Machine Learning para o reconhecimento molecular de entidades químicas com potencial farmacológico aplicada a SARS-CoV-2 (Covid-19)

1st **Rafael Vieira**
Instituto Federal de São Paulo - IFSP
Campinas, Brasil
ORCID: 0000-0001-9003-3209

2nd **Samuel Botter Martins**
Instituto Federal de São Paulo - IFSP
Campinas, Brasil
samuel.martins@ifsp.edu.br

***Resumo*: Este trabalho demonstra a pertinência das técnicas de *Machine Learning* (Aprendizagem de Máquina) na mineração e exploração racional de entidades químicas, a partir de produtos naturais de diferentes origens biológicas, além da prospecção de bioativos de interesse farmacológico por docagem molecular. A investigação apresentou potencial para descobertas inovadoras no que concerne às análises *in silico*, mediante o entrelaçamento entre algoritmos supervisionados e plataformas consolidadas de armazenamento de informações sobre moléculas (bancos moleculares). Para mais, será apresentada uma estratégia para reconhecimento de similaridade e das relações moleculares entre as entidades químicas isoladas de plantas e fungos e bactérias perante moléculas-fármacos.**

Palavras-chave: Produtos Naturais; Bancos Moleculares; mineração molecular; Quiminformática, Docking.

1. INTRODUÇÃO

As fontes naturais impulsionaram os estágios iniciais da Química Medicinal e da descoberta de medicamentos, produzindo valiosos agentes terapêuticos em uso até hoje [1]. Exemplos proeminentes de medicamentos oriundos de produtos naturais e que foram aprovados para uso clínico incluem, mas não estão limitados a penicilina, pilocarpina, reserpina e ácido salicílico [2]. As coleções de compostos naturais representam recurso crucial para manter, pesquisar, minerar e compartilhar informações químicas de diferentes moléculas.

Com o intuito de impulsionar ou até mesmo projetar os estudos multidisciplinares com produtos naturais, bancos moleculares foram criados para o compartilhamento de informações, surgindo como novos meios para acessar regiões desconhecidas do espaço terapêutico e químico [3].

Atualmente, existem vários bancos de dados de compostos naturais que permitem o armazenamento e o compartilhamento de dados para a triagem biológica de moléculas com distintas aplicações [4]. Recursos representativos a esse respeito são ChEMBL, PubChem e Binding Database, revisados coletivamente por Nicola et al. (2012) [5]. Há cerca de seis anos, havia aproximadamente cinco bancos de dados químicos públicos disponíveis, contendo entre 560 e 89.000 moléculas. Hoje, muitos mais bancos estão disponíveis e somam mais de 250.000 produtos naturais de domínio público, conforme revisado no relatório feito por Chen et al. (2018) [6] . Um número significativo de recursos de produtos naturais é construído e mantido por grupos acadêmicos e iniciativas sem fins lucrativos [7].

Diante da considerável ampliação dos bancos moleculares nos últimos anos, e à medida que a quantidade e a qualidade dos dados gerados sobre produtos naturais se expandem, surge a necessidade do uso de estratégias para a exploração mais eficiente de tais informações. Nesta perspectiva, tem se consolidado o potencial de aplicação de abordagens analíticas de inteligência artificial, tais como o Machine Learning (ML) (Aprendizado de Máquina), para a varredura, processamento e análise dos dados depositados nos bancos moleculares [8].

O ML é atualmente um dos tópicos mais importantes e em rápida evolução na descoberta de novas moléculas auxiliadas por computador. Em contraste com os modelos físicos que dependem de equações físicas explícitas, como Química Quântica ou simulações de dinâmica molecular, as abordagens de ML usam algoritmos de reconhecimento de padrões para discernir relações matemáticas entre observações empíricas de pequenas moléculas com o intuito de extrapolá-las para prever propriedades químicas, biológicas e físicas de novos compostos [9], permitindo a aplicação de descritores químicos em uma variedade de modelos de ML podendo predizer a variabilidade metabólica presente em uma amostra biológica [10], [11].

À vista disso, essa pesquisa permeou discussões de um tema emergente na literatura técnica-científica ao possibilitar o entrelaçamento de ML e Big Data para a mineração de estruturas químicas confiáveis, já depositadas em bancos moleculares públicos, e a construção de modelos capazes de processar dados em alto volume, velocidade, acurácia e com grande versatilidade na exploração de produtos naturais de diferentes origens.

Para mais, demonstrou a pertinência da ML na prospecção de moléculas-fármacos ou ainda fármaco-similares, como ferramenta do inventário da diversidade metabólica dos produtos naturais de origem biológica, e na formação de uma base de dados concreta, que forneça subsídios para o fomento de iniciativas farmacológicas, como ação em proteínas do SARS-CoV-2, por exemplo.

2. METODOLOGIA

2.1 - Prospecção de plataformas para o processo de mineração molecular

No mundo químico, há muita informação estrutural, formando assim um Big Data molecular, sendo listados, na última estimativa realizada em 2017, pelo menos 25 bancos [12]. No entanto, essas bases de dados moleculares ainda não são padronizadas em termos de informações disponíveis, principalmente no que se refere à classificação orgânica das moléculas. Em contrapartida, tais dados moleculares atendem aos princípios 5 V' s da designação de *Big Data*: volume,

velocidade, variedade, veracidade e valor, uma vez que passam por um processo de curadoria por instituições sérias e consolidadas [13]. Dito isso, a prospecção de plataformas moleculares candidatas à implementação de algoritmos de ML seguirá os seguintes critérios de seleção: a) plataformas com considerável impacto para o estado da arte em Química de Produtos Naturais; b) plataformas com acesso aberto e c) plataformas com relevante acervo.

2.2 - Uso de ML para a mineração de moléculas depositadas em plataformas moleculares de acesso aberto

Para a mineração de moléculas idênticas, análogas ou similares, depositadas em diferentes plataformas moleculares de acesso aberto, utilizou-se ferramentas de algoritmos ML supervisionados que permitiram a realização de estudos classificatórios de reconhecimento de padrões de similaridade, análises estatísticas para as moléculas apontadas como semelhantes ou análogas e o direcionamento dos estudos, para distinguir estruturas químicas isoladas de diferentes amostras biológicas. Através das plataformas Zinc15 (www.zinc15.docking.org) e Cortellis (www.cortellis.com) foram obtidas 22.154 moléculas, que foram separadas igualmente, formando, assim, a classe de produtos naturais e a classe de moléculas-fármaco. Ambas as plataformas foram acessadas em 15 de janeiro de 2023. Para tal, foram empregadas as avaliações métricas de precisão, recall, especificidade e F1 Score para a interpretação da matriz de confusão gerada e para a validação dos modelos previstos pelos algoritmos.

2.3 – Descritores moleculares (features) e Docagem molecular (triagem computacional)

As moléculas obtidas dos bancos de dados estavam representadas por uma string, chamada de SMILES (***S**implified **M**olecular-**I**nput **L**ine-**E**ntry **S**ystem*), e a partir desta representação, calculou-se 120 descritores químicos diversos, que por técnicas de *feature selection*, definiu-se 30 deles como os mais significativos para o modelo de *machine learning*. Tais descritores podem ser visualizados na figura 1.

Além disso, os metabólitos classificados pelos algoritmos de ML, como moléculas similares, ou seja, moléculas de produtos naturais que foram sinalizadas pelo algoritmo como candidatas a fármacos (falsos positivos) serão avaliados quanto ao seu potencial contra o coronavírus. Inicialmente, as moléculas selecionadas serão filtradas a partir da metodologia Lipinski [14], baseada na regra dos cinco, que afirma que uma molécula para ser considerado bom fármaco deverá ter: peso molecular (ExactMW) igual ou inferior a 500 Dalton - ExactMW <= 500; não mais do que 5 doadores de hidrogênio (NumHBD) - NumHBA <=5; não mais que 10 receptores de hidrogênio (NumHBA) - NumHBA <=10 e coeficiente de partição octanol-água log P (SlogP) menor ou igual 5 – SlogP <=5. O cálculo de tais parâmetros das moléculas foi obtido por meio da biblioteca RDKIT, na linguagem Python.

Ademais, definiu-se a separação entre os dados de treino e teste de acordo com a proporção 80/20. Além de pré-processamento dos dados, em que se pautou na exclusão de dados faltantes, e também a eliminação de duplicatas. A validação cruzada dos dados foi aplicada utilizando 10 folds. Além disso, buscou-se utilizar um método de padronização dos dados, e definiu-se o algoritmo de Robust Scaler para realizar tal procedimento.

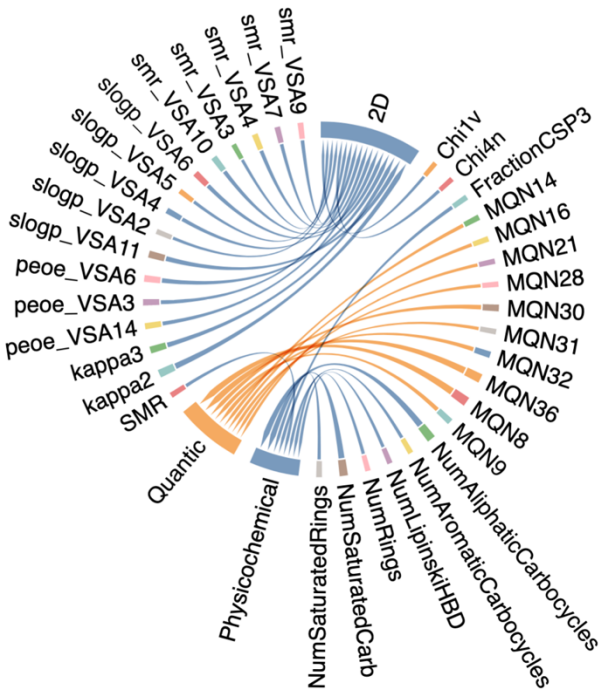


Figura 1 – Representação dos descritores químicos utilizados como features dos modelos de Machine Learning

A seleção da proteína baseou-se nos trabalhos divulgados por Yan e colaboradores [15], em que são apresentadas estruturas de microscopia crioeletrônica do receptor celular do coronavírus da síndrome respiratória aguda grave (SARS-CoV) e do novo coronavírus (SARS-CoV-2). A síndrome respiratória aguda grave-coronavírus 2 (SARS-CoV-2) é um vírus de RNA (ácido ribonucleico) que causa síndrome respiratória severa em humanos. O surto resultante da doença de coronavírus 2019 (COVID-19) emergiu como uma epidemia grave, ceifando mais de 696.742 vidas no Brasil, de acordo com dados do Sistema Único de Saúde – SUS (https://covid.saude.gov.br).

A estrutura desta proteína foi obtida da plataforma *Protein DataBase* (PDB) (www.rcsb.org) e está registrada na base de dados como 6M17. Por fim, as anotações moleculares, sinalizadas como fármaco-similares, foram empregadas no estudo de docagem molecular, utilizando o programa AutoDock4.0 [16], que emprega um conjunto de ferramentas para a predição da melhor conformação de um ligante de menor energia de interação. Na docagem molecular serão analisadas as interações entre as moléculas-fármacos contra a proteína "*spike*" utilizada pelo SARS-CoV-2 para infectar células humanas.

3. RESULTADOS E DISCUSSÃO

3.1 – Modelos de machine learning

Inicialmente, 10 modelos de aprendizado de máquina foram selecionados para realizar a classificação das moléculas entre produtos naturais e moléculas-fármacos. O

desempenho de cada modelo pode ser consultado na figura 2, evidenciando que o algoritmo de Random Forest foi o que proporcionou maiores métricas para a classificação molecular proposta, sendo, portanto, o modelo selecionado para realizar as predições com os dados de teste.

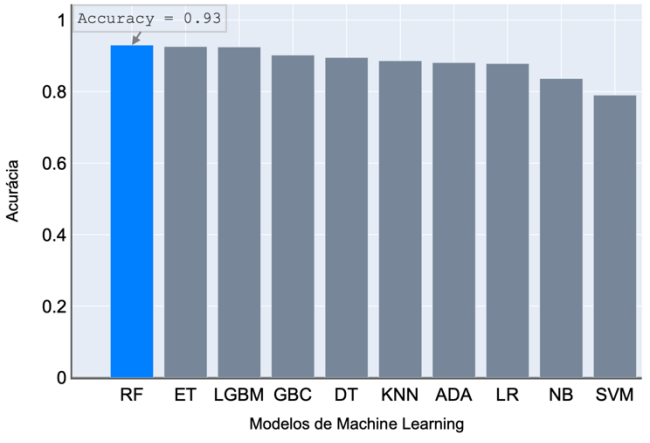


Figura 2 – Representação das métricas de avaliação para os 10 modelos de machine learning propostos para separação molecular entre produtos naturais e moléculas-fármacos. **RF** – Random Forest. **ET** – Extra Trees Classifier. **LGBM** - Light Gradient Boosting Machine. **GBC** - Gradient Boosting Classifier. **DT** - Decision Tree Classifier. **KNN** - K Neighbors Classifier. **ADA** - Ada Boost Classifier. **LR** - Logistic Regression. **NB** - Naive Bayes. **SVM** - SVM - Linear Kernel.

A partir da matriz de confusão (figura 3-A) é possível visualizar a distribuição das classificações, comparando os rótulos reais, com os valores preditos pelo modelo. Assumindo que o valor zero (0) foi atribuído às moléculas da classe "moléculas-fármaco", e o valor um (1) para as estruturas químicas de "produtos naturais", observa-se que das 4407 moléculas disponíveis no conjunto de teste, 4159 foram corretamente apontadas como pertencentes à sua classe real. Por outro lado, o modelo classificou de maneira errônea algumas estruturas químicas, gerando os grupos de falsos positivos, contendo 147 estruturas, que originalmente são fármacos (classe negativa), mas que apresentam características de produtos naturais (classe positiva), e também o grupo dos falsos negativos, indicando 101 moléculas de produtos naturais como tendo padrões de fármacos.

Relacionando tais observações, é possível gerar a curva ROC (figura 3-B), indicando a taxa de falsos positivos, com a taxa de verdadeiros positivos, assumindo que o modelo de Random Forest é um bom classificador para este tipo de problema. As métricas de avaliação do modelo selecionado pode ser consultadas na tabela 1. Nota-se que que a acurácia proporcionada pelo modelo foi elevada (92,99%), indicando a capacidade do algoritmo em reconhecer padrões distintos entre as duas classes estudadas, dando indicativos de que os descritores químicos utilizados são adequados para realizar a classificação assertiva entre as duas classes, além de evidenciar que produtos naturais e moléculas já consideradas fármacos apresentam feições moleculares distintas.

Para este trabalho, assumiu-se que o grupo de falsos negativos seria o espaço químico explorado, uma vez que trata-se da representação de estruturas de produtos naturais que apresentam os mesmos padrões de moléculas que já atuam em alvos biomacromoleculares de determinadas doenças.

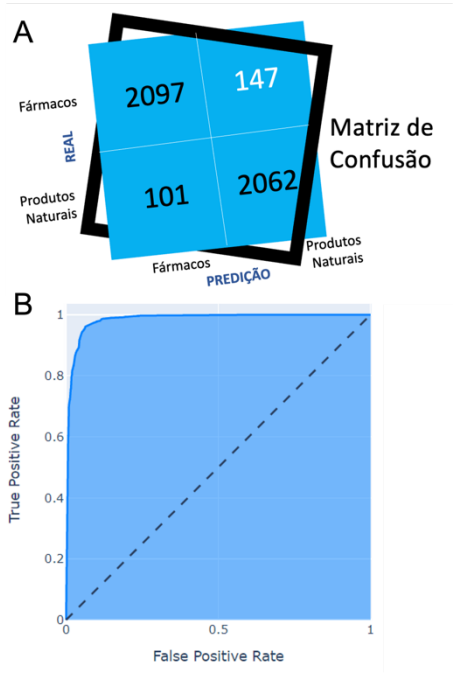


Figura 3 – Matriz de confusão para classificação das moléculas de produtos naturais e moléculas-fármacos geradas pelo modelo de Random Forest (A) e curva ROC (B).

Tabela 1 – Métricas de avaliação do modelo de Random Forest utilizado para o processo classificatório

CLASSE	Random Forest Classifier			
	Precisão	***Recall***	***F1-Score***	***Suporte[a]***
Fármacos	0,9540	0,9345	0,9442	2244
Produtos Naturais	0,9335	0,9533	0,9433	2163

[a]Acurácia do modelo: 0,9299

Dessa forma, foram identificadas as 101 moléculas agrupadas como falsos negativos e a partir das representaçoes SMILES foram convertidas em estruturas tridimensionais para que pudessem ser inseridas em testes computacionais de docagem molecular. Essa abordagem computacional busca encontrar as melhores acomodações entre duas moléculas simulando assim o processo de reconhecimento molecular. A partir das conformações em que moléculas pode assumir forma-se um complexo chamado de proteína-ligante, que permite estimar a afinidade química por aquele alvo, de modo que possa priorizar a molécula que melhor se adeque àquele alvo, dando indícios de que poderiam ser uma candidata a fármaco para aquela determinada doença.

O valor entre o ligante e a proteína é medido em kcal/mol, e quanto mais negativo, maior a afinidade pelo alvo. Dessa forma, todas as moléculas foram testadas no alvo 6M17 e selecionou-se a candidata-molecular que apresentou o valor mais expressivo pelo sítio reacional da proteína (Figura 4), cuja fórmula molecular é $C_{46}H_{69}NO_{12}$, ou seja, uma estrutura química bastante complexa, com 9 anéis estruturais e átomos eletronegativos, como o oxigênio e nitrogênio, além de ser

contemplada com muitas ligações com rotações livres que permitem interações intermoleculares com os aminoácidos da proteína, acarretando, assim, em uma maior afinidade (-9.5 kcal/mol).

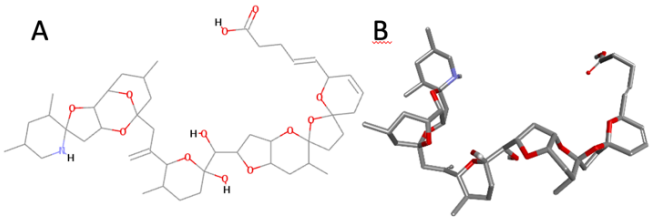


Figura 4 – Estrutura 2D (A) e 3D (B) da molécula ($C_{46}H_{69}NO_{12}$), com maior afinidade pelo alvo biomacromolecular da SARS-CoV-2

Por outro lado, neste conjunto de moléculas testadas no alvo biológico, há compostos que mesmo apresentando padrões de fármacos não conseguem modular a proteína estudada, como é o caso da estrutura de fórmula molecular $C_3H_80_2$. Seu arranjo conformacional não permite acomodação efetiva com o alvo, e faz com que a interação seja baixa, gerando apenas -2,8 kcal/mol no processo de complexação proteína-ligante. Informações como essas dão indicativos de que estruturas de produtos naturais com muitos átomos, ligações rotáveis e átomos com pares de elétrons livres, apresentam interação efetiva com esta proteína estudada. A figura 5 permite visualizar as interações do complexo formado pelo melhor ligante estudado.

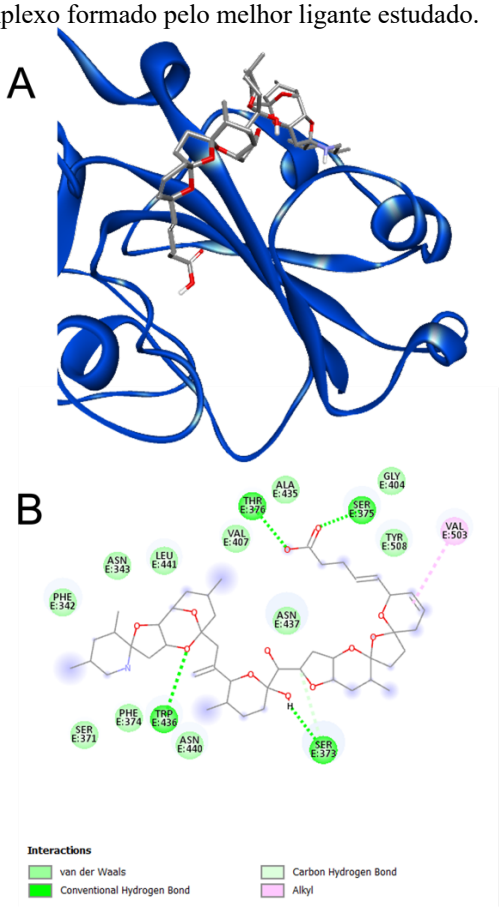


Figura 5 – Representação do complexo formado entre proteína-ligante (A) e interações moleculares entre aminoácidos e ligante (B) que apresentou maior efetividade na modulação estrutural do alvo biomacromolecular.

Analisando a figura 5-B é possível observar as interações entre alvo e ligante. As interações do tipo de van der Waals e ligações de hidrogênio predominam entre o complexo formado, que são complementadas com ligações carbono-hidrogênio e interações alquílicas comuns e pi-alquílicas, configurando a alta afinidade pelo alvo biomacromolecular. Uma vez que o ligante se liga nessa região, ele impede que a proteína do SARS-CoV-2 possam interagir com o receptor humano, impossibilitando a atuação do vírus de poder repassar seu material genético, e assim, causar infecção do trato respiratório.

4 . CONCLUSÃO

A crescente ampliação dos bancos moleculares nos últimos anos e a considerável expansão dos dados gerados sobre produtos naturais desencadearam o surgimento de estratégias para a exploração de tais informações. Neste contexto, surgiram as abordagens analíticas de inteligência artificial, tais como o *Machine Learning* auxiliando abordagens computacionais como a docagem molecular.

Assim, o que antes eram esforços isolados de pesquisa de produtos naturais, passou a envolver contribuições de equipes e comunidades. Os cientistas de produtos naturais não se dedicam mais exclusivamente ao isolamento e elucidação de estruturas. Eles também empreendem esforços para organizar e manter bancos de dados e melhorar as ferramentas para analisá-los. As mudanças na magnitude e no tipo de dados disponíveis aos pesquisadores na área de produtos naturais se refletem no que vemos hoje como um dos ramos da pesquisa de produtos naturais

Assim, o potencial de uma molécula como agente biológico, como no caso de moléculas fármaco-similares, pode ser investigado por protocolos de triagem virtual. Neste trabalho, utilizando o algoritmo de *Random Forest*, com acurácia de aproximadamente 93% na separação molecular entre produtos naturais e moléculas-fármacos, encontrou-se 147 moléculas de produtos naturais que foram apontadas como tendo padrões de fármacos, ou seja, falsos positivos, os quais foram transformados em estruturas químicas e encontrada a molécula $C_{46}H_{69}NO_{12}$ que apresenta potencial para atuar em alvos de SARS-CoV-2. Normalmente, um protocolo de triagem molecular envolve vários métodos em ordem consecutiva, tentando filtrar grandes bancos de dados para "escolher" os ligantes de interesse farmacológico. Até o momento, a docagem molecular, como estratégia de triagem virtual, foi aplicada com sucesso para identificar compostos de sucesso que geralmente são otimizados posteriormente

A pesquisa de produtos naturais, impulsionada por métodos computacionais, tem ainda limitações técnicas e de algoritmos, o que faz com que um número crescente de fontes de dados e a variedade de metabólitos permaneçam inexplorados. A disponibilidade de tecnologia levou ao desenvolvimento e implementação de uma infinidade de algoritmos que vão desde a coleta de dados até o perfilamento *in silico* e a triagem de produtos naturais utilizando inteligência artificial e docagem molecular. Assim, as perspectivas neste campo dizem respeito à construção e otimização de bancos de dados adequados para expansão do espaço químico, e assim, melhorar as campanhas baseadas em aprendizado de máquina em química de produtos naturais.

REFERÊNCIAS

[1] J. A. Beutler, "Natural products as a foundation for drug discovery", *Curr Protoc Pharmacol*, vol. 46, nº 1, p. 9–11, 2009.

[2] D. A. Dias, S. Urban, e U. Roessner, "A historical overview of natural products in drug discovery", *Metabolites*, vol. 2, nº 2, p. 303–336, 2012.

[3] P. Maragakis, H. Nisonoff, B. Cole, e D. E. Shaw, "A deep-learning view of chemical space designed to facilitate drug discovery", *J Chem Inf Model*, vol. 60, nº 10, p. 4487–4496, 2020.

[4] J. Zhang, J. Chen, Z. Liang, e C. Zhao, "New lignans and their biological activities", *Chem Biodivers*, vol. 11, nº 1, p. 1–54, 2014.

[5] G. Nicola, T. Liu, e M. K. Gilson, "Public domain databases for medicinal chemistry", *J Med Chem*, vol. 55, nº 16, p. 6987–7002, 2012.

[6] Y. Chen, M. de Lomana, N.-O. Friedrich, e J. Kirchmair, "Characterization of the chemical space of known and readily obtainable natural products", *J Chem Inf Model*, vol. 58, nº 8, p. 1518–1532, 2018.

[7] J. A. van Santen, S. A. Kautsar, M. H. Medema, e R. G. Linington, "Microbial natural product databases: moving forward in the multi-omics era", *Nat Prod Rep*, vol. 38, nº 1, p. 264–278, 2021.

[8] N. B. Cech, M. H. Medema, e J. Clardy, "Benefiting from big data in natural products: importance of preserving foundational skills and prioritizing data quality", *Nat Prod Rep*, vol. 38, nº 11, p. 1947–1953, 2021.

[9] J. Jeon, S. Kang, e H. U. Kim, "Predicting biochemical and physiological effects of natural products from molecular structures using machine learning", *Nat Prod Rep*, vol. 38, nº 11, p. 1954–1966, 2021.

[10] H. W. Kim *et al.*, "NPClassifier: A deep neural network-based structural classification tool for natural products", *J Nat Prod*, vol. 84, nº 11, p. 2795–2807, 2021.

[11] A. Capecchi e J.-L. Reymond, "Classifying natural products from plants, fungi or bacteria using the COCONUT database and machine learning", *J Cheminform*, vol. 13, nº 1, p. 1–11, 2021.

[12] Y. Chen, C. de Bruyn Kops, e J. Kirchmair, "Data resources for the computer-guided discovery of bioactive natural products", *J Chem Inf Model*, vol. 57, nº 9, p. 2099–2111, 2017.

[13] J. Anuradha e others, "A brief introduction on Big Data 5Vs characteristics and Hadoop technology", *Procedia Comput Sci*, vol. 48, p. 319–324, 2015.

[14] M. D. Shultz, "Two decades under the influence of the rule of five and the changing properties of approved oral drugs: miniperspective", *J Med Chem*, vol. 62, nº 4, p. 1701–1714, 2018.

[15] R. Yan, Y. Zhang, Y. Li, L. Xia, Y. Guo, e Q. Zhou, "Structural basis for the recognition of SARS-CoV-2 by full-length human ACE2", *Science (1979)*, vol. 367, nº 6485, p. 1444–1448, 2020.

[16] R. Huey, G. M. Morris, e S. Forli, "Using AutoDock 4 and AutoDock vina with AutoDockTools: a tutorial", *The Scripps Research Institute Molecular Graphics Laboratory*, vol. 10550, p. 92037, 2012.

INVESTIGAÇÃO DE ESTRATÉGIAS QUALITATIVAS E QUANTITATIVAS PARA A AVALIAÇÃO DE TÉCNICAS DE EXPLICABILIDADE APLICADAS A MODELOS DE APRENDIZADO DE MÁQUINA

1º Giovanna Nascimento Antonieti
Instituto Federal de Educação, Ciência
e Tecnologia de São Paulo
- IFSP Câmpus Campinas
Campinas, Brasil
g.antonieti@aluno.ifsp.edu.br

2º Samuel Botter Martins
Instituto Federal de Educação, Ciência
e Tecnologia de São Paulo
- IFSP Câmpus Campinas
Campinas, Brasil
samuel.martins@ifsp.edu.br

***Abstract*—Nos últimos anos, a inteligência artificial tem desempenhado um papel chave em diversos setores da nossa vida cotidiana, como em setores bancários com a análise de crédito ou até mesmo no sistema prisional, com modelos para cálculo da probabilidade de reincidência. Em muitos casos, o processo de tomada de decisão não é transparente, o que ajuda as pessoas a não terem confiança nesses métodos. Com isso, um crescente número de pesquisadores identificaram a necessidade de tornar esses modelos opacos, com processo de decisão ininteligível, mais compreensíveis. Dessa forma, temos visto um aumento no número de trabalhos na área de Inteligência Artificial Explicável (do inglês, *Explainable Artificial Intelligence* - XAI), que busca preencher essa lacuna por transparência no processo de decisão dos modelos. Entretanto, não há um consenso de como avaliar o quão boa está uma explicação fornecida por um método XAI, dificultando, assim, comparar essas técnicas. Em um esforço para identificar as principais técnicas de avaliação de métodos de XAI, realizamos uma revisão e categorização dos principais trabalhos nessa área, além de discutir os conceitos chaves para o campo de XAI. Com base em nossa pesquisa, concluímos com algumas sugestões de futuras direções de pesquisa para Inteligência Artificial Explicável.**

***Index Terms*—aprendizado de máquina, inteligência artificial explicável, interpretabilidade, explicabilidade, compreensibilidade**

I. Introdução

Inteligência artificial (IA) está no centro de diversos setores que adotaram novas tecnologias da informação [1] e embora suas raízes remontem a várias décadas atrás, com pesquisas iniciadas na década de 1950 [2]. Atualmente, há um consenso sobre a importância de que os sistemas inteligentes sejam dotados de capacidade de aprendizado e adaptação [3]. É por ter essas capacidades que os métodos de IA têm demonstrado seu potencial para revolucionar diversas esferas econômicas, alcançando níveis de desempenho sem precedentes ao aprender a resolver situações cada vez mais complexas. Por essas razões, os modelos de IA tem se tornado fundamentais para o desenvolvimento da sociedade [4], e, por vezes, superando o desempenho humano para uma variedade de problemas [5], [6].

Ao passo que os primeiros sistemas de IA eram facilmente interpretáveis e compreensíveis, recentemente temos visto uma ascensão de sistemas de decisão opacos, ou seja, sistemas em que o seu funcionamento não é compreensível para o usuário, como, por exemplo, as redes neurais profundas (DNNs, do inglês *Deep Neural Network*), que combina centenas de camadas e milhões de parâmetros [3]. Essa opacidade criou a necessidade de arquiteturas de Inteligência Artificial Explicável (XAI, do inglês *eXplainable Artificial Intelligence*) motivadas por três razões principais de acordo com Fox *et al.* [7] e Dovsilovic *et al.* [8]: (1) a necessidade de criar modelos mais transparentes; (2) a necessidade de técnicas que permitam que humanos interajam com elas; e (3) o requisito de confiabilidade nas suas inferências.

A partir de 2016, explicabilidade tem sido identificada como o fator chave para adoção de sistemas inteligentes em contextos mais amplos [9], [10], o que fez com que o tema venha recebendo bastante atenção da academia [11], [12]. O aumento do foco no tópico é resultado direto do uso de métodos de aprendizado de máquina no nosso cotidiano e seu impacto em processos críticos de tomada de decisão, sem serem capazes de prover informações detalhadas sobre o processo de decisão ou predição [13]. Entretanto, é importante salientar que as definições de técnicas de XAI são genéricas e por vezes não há um consenso sobre sua definição [14], o que pode explicar, ao menos parcialmente, porque os métodos de explicabilidade são tão distintos. Segundo Gunning *et al.* [15], a inteligência artificial explicável pretende produzir modelos inteligíveis, enquanto mantém um alto nível de desempenho

de aprendizado, e permita que usuários possam entender, confiar, e efetivamente administrar a geração emergente de sistemas artificialmente inteligentes. Em contrapartida, para Adadi e Berrada [11], a inteligência artificial explicável se refere a movimentos, iniciativas e esforços feitos em resposta às preocupações de transparência e confiança em sistemas IA, mais do que a uma técnica formal.

Diversos novos métodos são propostos e descobertas são compartilhadas sobre interpretabilidade e explicabilidade, o que nos leva a uma produção abundante de conhecimento, porém ainda muito dispersas, em que cada autor segue uma linha diferente de pesquisa, utilizando diferentes categorias de métodos de explicabilidade, e por vezes não há uma padronização na fase de avaliação desses métodos. As métricas de XAI são realmente necessárias para apoiar avaliação das técnicas e ferramentas existentes, já propostas pela comunidade, e deveriam avaliar a qualidade da explicação, assim como o seu impacto no desempenho do modelo e na confiança do usuário [3]. Embora temos diversas técnicas de XAI sendo propostas na literatura, não vemos o mesmo acontecendo com as métricas para avaliar e comparar essas técnicas. O que nos leva a um problema na comparação de métodos por falta de padronização ou de conhecimento de qual técnica avaliativa utilizar, de acordo com o método de explicabilidade empregado.

Nessa pesquisa identificamos e categorizamos os principais trabalhos na área de métricas presentes na literatura para avaliação dos mecanismos de explicabilidade quando aplicadas a modelos de aprendizado de máquina. Observamos que é possível agrupar esses trabalhos em três grandes áreas: trabalho de avaliação qualitativa, trabalho de avaliação quantitativa e trabalhos de pesquisa, que reúnem outras pesquisas e as utilizam para propor seu próprio método de avaliação.

II. Terminologia

Ser capaz de fornecer uma explicação do porquê certa decisão foi tomada tem se tornado uma característica desejável de sistemas inteligentes [16]. Muitos trabalhos introduzem conceitos chaves para a área de XAI como *transparência, interpretabilidade, compreensibilidade e explicabilidade*, a despeito da falta de consenso sobre esses termos permanecer um problema. O termo explicabilidade por vezes é utilizado como sinônimo de interpretabilidade, normalmente na comunidade de IA, enquanto a comunidade de engenharia de *software* da preferência pelo termo compreensibilidade [17]. O termo *explainable artificial intelligence* foi cunhado pela primeira vez em 2004 por Van *et al.* [18], para descrever a habilidade de seus sistemas explicarem o comportamento de entidades controladas por IA em aplicações de jogos de simulação, sendo a primeira noção de XAI subsidiada por sistemas especialistas na segunda metade da década de 1980 [19], [20]. As definições adotadas no decorrer desse trabalho podem ser encontradas na Tabela I.

Segundo Verhagen *et al.* [21], a principal diferença entre transparência e explicabilidade se resume a divulgar versus esclarecer. A transparência visa fornecer respostas descritivas através de conhecimento sobre elementos do sistema. Já a explicabilidade visa facilitar a compreensão, esclarecendo as relações entre os elementos do sistema. Ainda de acordo com Verhagen *et al.* [21], a diferença fundamental entre interpretabilidade e compreensibilidade está relacionada a uma diferença no esforço cognitivo necessário para se ter o conhecimento do sistema. A interpretabilidade requer mais esforço cognitivo porque implica em inferir o significado e as relações entre as informações, sem que o conhecimento desses significados ou as relações entre eles esteja explícito. A compreensibilidade, por sua vez, requer menos esforço cognitivo porque implica em conhecer o significado e as relações divulgadas e esclarecidas.

Na literatura temos diversos trabalhos que propõe uma taxonomia para as técnicas de XAI. Fan *et al.* [24] propõem uma divisão entre análise de explicabilidade *post-hoc* e modelagem interpretável *ad-hoc*, que busca construir modelos interpretáveis. A análise de explicabilidade *post-hoc* explica modelos que já existem e esses métodos podem ser agrupados nas seguintes categorias:

- *Análise de feature* são técnicas centradas em comparar, analisar e visualizar a importância de cada *feature* ao modelo, permitindo encontrar *features* sensíveis e formas de processá-las, de tal maneira que a lógica do modelo possa ser explicada;
- *Inspeção do modelo* são métodos que usam algoritmos externos para extrair de forma sistemática estruturas importantes e informações paramétricas dos mecanismos internos do modelo;
- *Métodos de saliência* identificam atributos dos dados de entrada que são mais relevantes para a predição ou são uma representação latente do modelo;
- *Proxy*, nessa categoria os métodos constroem modelos mais simples e interpretáveis que possuem grande semelhança com o modelo caixa-preta treinado;
- *Análise física/matemática avançada* colocam o modelo em um quadro teórico matemático, no qual os mecanismos da rede podem ser entendidos através de ferramentas matemáticas avançadas;
- *Explicação por caso* prove exemplos representativos que capturam a essência de um modelo;
- *Explicação textual* geram um texto descritivo em tarefas de imagem-texto que são condutoras para entender o comportamento do modelo.

III. Metodologia

O presente trabalho constitui-se de uma revisão de literatura, buscando a identificação de métricas para avaliação das técnicas de explicabilidade e possíveis lacunas nesse campo de conhecimento. Para a compilação dos resultados, realizamos as seguintes fases:

1ª fase: **Levantamento dos trabalhos a serem analisados através de uma revisão do estado da arte utilizando-se o *Google Scholar***. O termo de busca utilizado foi “measuring interpretability

Tabela I: Resumo das definição adotadas.

Termo	Definição
Interpretabilidade	Segundo Arrieta *et al.* [3], é a habilidade de explicar ou fornecer significado em termos humanamente compreensíveis. Dessa forma, podemos entender que a interpretabilidade é uma característica passiva do modelo que é adicionada durante o desenvolvimento de sua arquitetura.
Compreensibilidade	Se refere ao nível em que os usuários tem conhecimento dos elementos do sistema divulgados e esclarecidos, e as relações e dependências entre eles [21].
Explicabilidade	Esta característica está associada com a ideia da explicação ser uma interface entre humanos e o modelo preditivo ou tomador de decisões [22], onde temos uma relação dos valores dos atributos sendo estabelecida com a predição do modelo de forma humanamente inteligível. Dessa forma, percebemos que a explicabilidade se refere a uma característica *ativa* do modelo, já que descreve o processo realizado e esclarece o funcionamento interno do modelo.
Transparência do sistema	Representa a capacidade do modelo de divulgar os elementos relevantes do sistema para os usuários, permitindo que eles acessem, analisem e explorem essas informações divulgadas [21].
Modelo caixa-preta	Um modelo é considerado caixa-preta se o mapeamento entre seus parâmetros e a saída é escondido dos usuários [23].
Inteligência artificial explicável	É definida como uma entidade que fornece detalhes do seu funcionamento de forma facilmente compreensível [3]. É importante ressaltar que o nível de detalhe fornecido é variável, dependendo do público que o receberá, se são pessoas que possuem conhecimento prévio do assunto ou não.

explainable_artificial_intelligence -medical -health -healthy" – a remoção do termo *medical* deve-se à tentativa de se ampliar os resultados da busca. Essa consulta retornou cerca de 3.800 trabalhos, mas observando as páginas se tornou claro que apenas as primeiras dez páginas, com dez resultados cada, continham trabalhos relevantes. Um ponto importante a ser ressaltado nessa etapa é: a busca foi ordenada por trabalhos recentes, com ano de publicação a partir de 2021, uma vez que é uma área de pesquisa relativamente nova.

2ª fase: **Filtragem dos trabalhos que aparecem nessas dez primeiras páginas**. Realizamos esta etapa com o objetivo de excluir os trabalhos que não tratam da avaliação ou mensuração da explicabilidade dos sistemas inteligentes. Durante a filtragem também foram adicionados alguns trabalhos muito citados por aqueles que aparecem na busca.

3ª fase: **Categorização dos trabalhos** em:

- Trabalhos que aparecem nas buscas e falam primariamente sobre as métricas/formas para avaliação da explicabilidade dos sistemas;
- Pesquisas sobre métricas que são citadas nos trabalhos que aparecem na busca;
- Outros trabalhos.

Apenas os trabalhos que se encaixam nas duas primeiras categorias foram lidos e analisados para essa pesquisa. Dessa forma, selecionamos e analisamos um total de 21 artigos. A próxima seção apresenta os resultados de nossa investigação.

IV. Resultados

Analisando os trabalhos que tem como tema central formas de avaliar técnicas de explicabilidade, pudemos observar uma clara distinção entre os que proprõem métricas de avaliação, ou seja, uma forma de avaliar quantitativamente esses métodos de explicabilidade, os trabalhos que apresentam formas qualitativas de avaliar tais técnicas e os trabalhos de pesquisa *surveys*, que buscam reunir os resultados presentes na literatura para muitas vezes recomendar uma nova forma de avaliação.

Separamos nossa análise em três diferentes vertentes, como mostradas nas seções a seguir.

A. Avaliação qualitativa

Alguns trabalhos propõem estratégias para avaliar as técnicas de XAI qualitativamente, avaliando alguns pontos desejados nos resultados e técnicas, outros categorizam as formas de avaliação de acordo com a audiência ou tipo da análise utilizada. Por exemplo, Schlgel *et al.* [25] determinam uma maneira de validar as explicações de séries temporais por meio da adição de perturbações nos dados e avaliando o comportamento das explicações geradas. Enquanto em um dos trabalhos mais relevantes da área [10], Doshi-Velez e Kim propõe uma taxonomia das técnicas de avaliação, categorizando-as em três tipos: (1) a avaliação baseada na aplicação, que envolve conduzir experimentos com humanos em tarefas reais e avaliar a qualidade da explicação no contexto da tarefa final; (2) a avaliação baseada em humanos, que são experimentos conduzidos em tarefas mais simples mas mantém a essência da aplicação alvo; e (3) a avaliação baseada em funcionalidade, que contemplam experimentos que não necessitam de humanos, além de utilizar uma definição formal de interpretabilidade como *proxy* para medir a qualidade da explicação gerada.

Em [21], os autores definem algumas premissas dos sistemas XAI, requisitos para alcançá-las e como validá-las, essas exigências são:

- *A explicabilidade do sistema resulta em mais conhecimento/modelos mentais completos do sistema do que transparência.* Nesse caso a implementação da

transparência do sistema e a medição do conhecimento do usuário sobre o sistema seriam as condições necessárias pra se obter essa reinvindicação. Já para a validação, seria importante medir o conhecimento e compreensão do usuário sobre o sistema, seja de forma subjetiva ou objetiva, como por exemplo através de perguntas sobre o sistema e a escolha de qual das saídas possíveis é a de maior qualidade.

- *O aumento do conhecimento do usuário de um sistema resulta em melhorias na colaboração do agente e, eventualmente, no desempenho da equipe*. Medidas de colaboração podem ajudar na validação desse requisito.
- *A transparência do sistema já permite a controlabilidade e direcionabilidade do sistema, mas não a contestabilidade, previsibilidade, verificabilidade e rastreabilidade do sistema.* A implementação da transparência do sistema e medição da controlabilidade, previsibilidade, verificabilidade e rastreabilidade do sistema podem contribuir a alcançar e validar esse ponto.
- *A explicabilidade do sistema permite contestabilidade, previsibilidade, verificabilidade e rastreabilidade.* A adição de explicabilidade ao sistema, além da medição da contestabilidade, previsibilidade, verificabilidade e rastreabilidade do sistema podem contribuir na busca por essa cláusula.

Alguns outros trabalhos seguem uma linha diferente, como é o caso de Hoffman *et al.* [26] que propõem a escala de satisfação da explicação composta por sete itens do tipo *Likert*, escala de resposta composta de cinco opções onde os perguntados especificam seu nível de concordância com uma afirmação, sendo esses itens referentes as explicações: compreensibilidade, satisfação, detalhe, precisão, completude, usabilidade, utilidade e confiabilidade. A recomendação de uso feita pelos autores é utilizar essa escala para avaliar as explicações produzidas por qualquer sistema XAI, sendo os principais usuários os próprios pesquisadores de IA. Combs *et al.* [27] baseam-se nos dez princípios heurísticos de Nielsen para criar os tópicos de avaliação do modelo XAI, princípios esses normalmente utilizados para a avaliação da usabilidade de um interface.

Todos os trabalhos apresentados neste tópico, com excessão de [25], podem ser aplicados a diferentes técnicas de explicabilidade, desde que o resultado dessas técnicas seja compreensível para o usuário final, uma vez que não dependem da forma da saída do modelo e sim da capacidade do usuário de compreendê-la e avaliá-la.

B. Avaliação quantitiva

A fim de avaliar as técnicas de XAI de forma quantitativa, alguns trabalhos se propõe a formular ou utilizar algumas métricas que avaliam diferentes pontos desses métodos de explicabilidade.

Alguns autores citam métricas difíceis de mensurar, como Islam *et al.* [28] que criam uma métrica baseada em *chunks* cognitivos, pedaços de informação, contidos na representação da explicação. Outros optam por medidas mais simples como Dam *et al.* [29] que propõem utilizar o tamanho do modelo gerado pelo método de XAI, no caso de métodos de *proxy*, como uma métrica para avaliar a explicabilidade da técnica.

Em [30], são propostas três métricas focadas em avaliar mapas de saliência, sendo necessário para essa métrica delimitar uma área de importância, chamada de área de *trigger*, na imagem original, que pode ser delimitada com base nos resultados de métodos de explicabilidade de mapas de saliência, dessa forma as métricas são calculadas em cima dessas regiões, sendo as elas:

- *Intersecção sobre união* que mede a sobreposição entre a área de *trigger original* e a área detectada dividida pela união dessas áreas. Essa métrica varia entre 0-1 e quanto mais alta melhor foi a detecção da área de *trigger* pelo método XAI;
- *Taxa de recuperação* mede a porcentagem média de imagens recuperadas classificadas corretamente, para isso as imagens possuem uma adição de perturbação nas regiões de *trigger*. Quanto maior a taxa, mais eficiente foi a recuperação dessa região, o que por sua vez no leva a concluir uma melhor detecção da área;
- *Diferença de recuperação* que mede a diferença entre a imagem recuperada e a imagem original, quanto menor seu valor significa que o método de explicabilidade ajudou efetivamente a identificar o *trigger* para removê-lo;
- *Tempo computacional*, apesar de não ser uma métrica proposta pelos autores, é citado como uma métrica para avaliar a explicabilidade, definida pelo tempo médio de execução utilizado por um método XAI para gerar o mapa de saliência.

Em [31], os autores utilizam técnicas de explicabilidade para gerar um vetor de *features* fortes e com base nesse vetor avaliam a técnica utilizada. Eles propõem duas métricas, chamadas de *Pontuação de explicabilidade forte* e *Pontuação de explicabilidade leve*. Tendo a primeira o objeto de avaliar se as *features* da imagem estão presente no vetor de *features* fortes, podendo apenas assumir valores de 0 ou 1, seria mais uma métrica boleana que apresenta se uma característica é ou não importante. A segunda busca avaliar a sobreposição da explicação estimada e o *groundtruth*. Sendo essa última principalmente aplicada a imagens.

Em [32] o autor determina quatro formas objetivas de analisar a efetividade do método XAI, são elas:

- *D*, uma forma de quantificar a diferença entre o desempenho de um agente caixa-preta e o melhor modelo transparente observado, o que pode ajudar a justificar a utilização de modelos opacos, podendo ser utilizados tanto em avaliações binárias como não binárias;
- *R* que representa o número de regras em um agente de explicação, útil para métodos transparentes;
- *F* é o número de *features* que foram utilizadas para construir a explicação;
- *S* mede a estabilidade de uma explicação.

Rosenfeld [32] acredita que essas métricas são comple-

mentares, sendo desejável que uma explicação transparente seja estável, tenha um desempenho semelhante ao modelo caixa preta e contenha poucas regras, tendo uma alta pontuação em D, R e S.

Veldhuis *et al.* [33] propõem uma nova métrica de realismo para explicações contrafactuais, que avalia a plausabilidade da combinação de suas *features*. Além disso também cita a interatividade, a esparcidade e a distância entre a explicação gerada e a amostra a ser explicada como formas de avaliação dos métodos de explicação por caso.

Em [34], os autores comparam técnicas de explicabilidade e atenção avaliando a concordância entre elas através da Correlção Kendall-τ, que verifica a semelhança entre a ordem dos dados, além de debaterem um pouco sobre a existência de uma explicação, muitas vezes considerada como a ideal.

C. Pesquisas na área

Gilpin *el at* [35] falam sobre duas formas de se avaliar uma explicação: através da sua interpretabilidade e de acordo com sua completude. O objetivo da interpretabilidade, segundo os autores, é descrever os componentes internos de um sistema de forma que seja humanamente compreensível. O sucesso desse objetivo está ligado à cognição, conhecimento e preconceitos do usuário. Enquanto o objetivo da completude é descrever a operação do sistema de forma precisa. Uma explicação é mais completa quando permite que o comportamento do sistema seja antecipado em um maior número de situações. Analisando esse trabalho podemos concluir que a maioria das pesquisas realiza um dos seguintes tipos de avaliação de suas explicações:

- *Completude em relação ao modelo original*. Um modelo *proxy* pode ser avaliado diretamente de acordo com o quanto ele se aproxima do modelo original que está sendo explicado;
- *Completude medida em uma tarefa substituta*. Algumas explicações não esclarecem diretamente as decisões de um modelo, mas sim algum outro atributo que pode ser avaliado;
- *Capacidade de detectar modelos com vieses*. Uma explicação que revela sensibilidade a um fenômeno específico pode ser testada quanto à sua capacidade de revelar modelos com a presença ou ausência de um viés relevante;
- *Avaliação humana*. Os humanos podem avaliar as explicações quanto à razoabilidade, ou seja, quão bem uma explicação corresponde às suas expectativas.

Mohseni *et al.* [36] criam uma taxonomia para os métodos e métricas de avaliação dos métodos XAI, baseada no usuário alvo e no tipo de avaliação. A maioria dos sistemas XAI é projetada exclusivamente para um tipo de explicação e um usuário específico, o que afeta a sua finalidade. De acordo com os autores, o primeiro passo para avaliar um sistema seria identificar o usuário alvo, e então escolher uma técnica XAI que beneficie os objetivos e necessidades para o usuário escolhido. De acordo com as escolhas realizadas, é possível escolher o tipo e o formato de explicação para continuar com o plano de projeto do sistema.

Em outro trabalho da área Mohseni *et al.* [37] criam uma categorização de acordo com os objetivos do projeto dos métodos de explicabilidade e as medidas de avaliação. Sendo os objetivos definidos pelo usuário final, onde cada categoria possui os seus objetivos específicos. Já as formas de avaliação são divididas em:

- *Mapa mental*. Seguindo as teorias da psicologia cognitiva, um modelo mental é uma representação de como os usuários entendem um sistema. Pesquisadores em IHC estudam os modelos mentais dos usuários para determinar sua compreensão de sistemas inteligentes em várias aplicações. No contexto do XAI, as explicações ajudam os usuários a criar um modelo mental de como a IA funciona. A análise de entrevistas, pensamentos em voz alta e autoexplicações dos usuários fornecem informações valiosas sobre os processos de pensamento e modelos mentais dos usuários.
- *Satisfação e utilidade*. Os pesquisadores usam diferentes medidas subjetivas e objetivas para compreensão, como utilidade e suficiência de detalhes para avaliar o valor explicativo para os usuários. Seus resultados mostraram uma forte relação entre a satisfação do usuário e a transparência percebida. Outra linha de pesquisa estuda se sistemas inteligíveis são sempre apreciados pelos usuários ou se tem um valor condicional.
- *Confiança do Usuário*. A confiança do usuário em um sistema inteligente é um fator afetivo e cognitivo que influencia as percepções positivas ou negativas de um sistema. A confiança inicial do usuário e o desenvolvimento da confiança ao longo do tempo foram estudados e apresentados com termos diferentes, como confiança rápida, confiança padrão e confiança suspeita. O conhecimento prévio e as crenças são importantes para moldar o estado inicial de confiança; no entanto, ela pode mudar em resposta à exploração e desafio do sistema com casos extremos. É essencial levar em consideração a experiência e o aprendizado do usuário ao longo do tempo ao trabalhar com sistemas complexos de IA.
- *Desempenho de Tarefa Humano-IA*. Um dos principais objetivos do XAI é ajudar os usuários finais a serem mais bem-sucedidos em suas tarefas envolvendo sistemas de aprendizado de máquina. Assim, o desempenho da tarefa de IA é uma medida relevante para todos os três tipos de usuários (novatos em IA, *experts* em dados e *experts* em IA).
- *Métricas computacionais*. As medidas computacionais são comuns no campo de aprendizado de máquina para avaliar a exatidão e a integridade das técnicas de interpretabilidade em termos de explicar o que o modelo aprendeu.

Mohseni *et al.* [37] também propõe um *framework* de avaliação baseado em camadas, começando na camada externa (metas do sistema XAI), abordando as necessidades do usuário

final na camada do meio (interface explicável) e focando nos algoritmos interpretáveis na camada mais interna.

Carvalho *et al.* [38] focam nos trabalhos que levantam os desideratos e axiomas desejados dos métodos e modelos de explicabilidade, levando em conta também trabalhos na área de ciência social, além de propor uma taxonomia para os métodos de explicabilidade.

V. Conclusão

Focamos esse trabalho na revisão de pesquisas voltadas a avaliação dos métodos de explicabilidade e suas explicações resultantes. Observamos que temos diversos artigos propondo diferentes técnicas de XAI, que não parecem levar em conta trabalhos anteriores, sem um consenso na definição de termos chaves da área. Essa variedade de técnicas também não ajuda na identificação de formas de avaliação, uma vez que cada método foca em um tipo de trabalho específico.

Durante o decorrer desse trabalho pudemos observar que há diferentes métricas propostas na literatura e a maioria aponta para uma explicação ideal, o que é um pressuposto questionável, mas muitas vezes considerado como fundamental nos paradigmas de avaliação. Pudemos constatar também que por vezes os autores tentam avaliar algo subjetivo, como é o caso do número de pedaços de informação que utilizamos para entender uma explicação.

Nossa visão é que, à medida que a comunidade aprender e avançar de forma colaborativa, combinando ideias de diferentes campos, o estado geral da XAI melhorará drasticamente, resultando em métodos que criam confiança em sistemas de aprendizado profundo e fornecem informações utilizáveis na operação profunda da rede, permitindo a compreensão e melhoria do comportamento do sistema. Algumas direções interessantes de pesquisa futura é a adoção de definições padronizadas para os termos chaves, além da definição clara dos objetivos de cada método XAI, esses dois pontos podem contribuir de forma significativa para as pesquisas de métodos de avaliação.

References

[1] S. Russell and P. Norvig, "Artificial intelligence: a modern approach," 2002.

[2] J. McCarthy, M. L. Minsky, N. Rochester, and C. E. Shannon, "A proposal for the dartmouth summer research project on artificial intelligence, august 31, 1955," *AI magazine*, vol. 27, no. 4, pp. 12–12, 2006.

[3] A. B. Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-López, D. Molina, R. Benjamins, *et al.*, "Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai," *Information fusion*, vol. 58, pp. 82–115, 2020.

[4] D. M. West, *The future of work: Robots, AI, and automation*. Brookings Institution Press, 2018.

[5] P. P. Angelov, E. A. Soares, R. Jiang, N. I. Arnold, and P. M. Atkinson, "Explainable artificial intelligence: an analytical review," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 11, no. 5, p. e1424, 2021.

[6] C. Meske, E. Bunde, J. Schneider, and M. Gersch, "Explainable artificial intelligence: objectives, stakeholders, and future research opportunities," *Information Systems Management*, vol. 39, no. 1, pp. 53–63, 2022.

[7] M. Fox, D. Long, and D. Magazzeni, "Explainable planning," in *2017 Workshop on Explainable Artificial Intelligence, XAI, Internacional Joint Conferences on Artificial Intelligence, Inc*, pp. 24–30, IJCAI, 2017.

[8] F. K. Došilović, M. Brčić, and N. Hlupić, "Explainable artificial intelligence: A survey," in *2018 41st International convention on information and communication technology, electronics and microelectronics (MIPRO)*, pp. 0210–0215, IEEE, 2018.

[9] M. T. Ribeiro, S. Singh, and C. Guestrin, "Model-agnostic interpretability of machine learning," *arXiv preprint arXiv:1606.05386*, 2016.

[10] F. Doshi-Velez and B. Kim, "Towards a rigorous science of interpretable machine learning," *arXiv preprint arXiv:1702.08608*, 2017.

[11] A. Adadi and M. Berrada, "Peeking inside the black-box: a survey on explainable artificial intelligence (xai)," *IEEE access*, vol. 6, pp. 52138–52160, 2018.

[12] A. Preece, "Asking 'why'in ai: Explainability of intelligent systems–perspectives and challenges," *Intelligent Systems in Accounting, Finance and Management*, vol. 25, no. 2, pp. 63–72, 2018.

[13] G. N. Antonieti, "Análise de métodos de explicabilidade de redes neurais profundas para a classificação de elsagate," 2021. Orientador: Sandra Eliza Fontes de Avila. 2021. 74p. Dissertação (Mestrado) – Ciência da Computação, Instituto de Computação, Universidade de Campinas, Campinas, 2021.

[14] C. Rudin, "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead," *Nature Machine Intelligence*, vol. 1, no. 5, pp. 206–215, 2019.

[15] D. Gunning, M. Stefik, J. Choi, T. Miller, S. Stumpf, and G.-Z. Yang, "Xai—explainable artificial intelligence," *Science Robotics*, vol. 4, no. 37, p. eaay7120, 2019.

[16] D. Doran, S. Schulz, and T. R. Besold, "What does explainable ai really mean? a new conceptualization of perspectives," *arXiv preprint arXiv:1710.00794*, 2017.

[17] J. M. Alonso, C. Castiello, and C. Mencar, "A bibliometric analysis of the explainable artificial intelligence research field," in *International conference on information processing and management of uncertainty in knowledge-based systems*, pp. 3–15, Springer, 2018.

[18] M. Van Lent, W. Fisher, and M. Mancuso, "An explainable artificial intelligence system for small-unit tactical behavior," in *Proceedings of the national conference on artificial intelligence*, pp. 900–907, Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2004.

[19] B. G. Buchanan and E. H. Shortliffe, "Rule-based expert systems: the mycin experiments of the stanford heuristic programming project," 1984.

[20] M. R. Wick and W. B. Thompson, "Reconstructive expert system explanation," *Artificial Intelligence*, vol. 54, no. 1-2, pp. 33–70, 1992.

[21] R. S. Verhagen, M. A. Neerincx, and M. L. Tielman, "A two-dimensional explanation framework to classify ai as incomprehensible, interpretable, or understandable," in *International Workshop on Explainable, Transparent Autonomous Agents and Multi-Agent Systems*, pp. 119–138, Springer, 2021.

[22] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi, "A survey of methods for explaining black box models," *ACM computing surveys (CSUR)*, vol. 51, no. 5, pp. 1–42, 2018.

[23] A. Das and P. Rad, "Opportunities and challenges in explainable artificial intelligence (xai): A survey," *arXiv preprint arXiv:2006.11371*, 2020.

[24] F.-L. Fan, J. Xiong, M. Li, and G. Wang, "On interpretability of artificial neural networks: A survey," *IEEE Transactions on Radiation and Plasma Medical Sciences*, vol. 5, no. 6, pp. 741–760, 2021.

[25] U. Schlegel, H. Arnout, M. El-Assady, D. Oelke, and D. A. Keim, "Towards a rigorous evaluation of xai methods on time series," in *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pp. 4197–4201, IEEE, 2019.

[26] R. R. Hoffman, G. Klein, and S. T. Mueller, "Explaining explanation for "explainable ai"," in *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 62, pp. 197–201, SAGE Publications Sage CA: Los Angeles, CA, 2018.

[27] K. Combs, M. Fendley, and T. Bihl, "A preliminary look at heuristic analysis for assessing artificial intelligence explainability," *WSEAS transactions on computer research, available at: https://doi.org/10.37394/232018.2020*, vol. 8, 2020.

[28] S. R. Islam, W. Eberle, and S. K. Ghafoor, "Towards quantification of explainability in explainable artificial intelligence methods," in *The thirty-third international flairs conference*, 2020.

[29] H. K. Dam, T. Tran, and A. Ghose, "Explainable software analytics," in *Proceedings of the 40th International Conference on Software Engineering: New Ideas and Emerging Results*, pp. 53–56, 2018.

[30] Y.-S. Lin, W.-C. Lee, and Z. B. Celik, "What do you see? evaluation of explainable artificial intelligence (xai) interpretability through neural backdoors," *arXiv preprint arXiv:2009.10639*, 2020.
[31] R. Machlev, M. Perl, J. Belikov, K. Y. Levy, and Y. Levron, "Measuring explainability and trustworthiness of power quality disturbances classifiers using xai—explainable artificial intelligence," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 8, pp. 5127–5137, 2021.
[32] A. Rosenfeld, "Better metrics for evaluating explainable artificial intelligence," in *Proceedings of the 20th international conference on autonomous agents and multiagent systems*, pp. 45–50, 2021.
[33] M. S. Veldhuis, S. Ariëns, R. J. Ypma, T. Abeel, and C. C. Benschop, "Explainable artificial intelligence in forensics: Realistic explanations for number of contributor predictions of dna profiles," *Forensic Science International: Genetics*, vol. 56, p. 102632, 2022.
[34] M. Neely, S. F. Schouten, M. Bleeker, and A. Lucic, "A song of (dis) agreement: Evaluating the evaluation of explainable artificial intelligence in natural language processing," *arXiv preprint arXiv:2205.04559*, 2022.
[35] L. H. Gilpin, D. Bau, B. Z. Yuan, A. Bajwa, M. Specter, and L. Kagal, "Explaining explanations: An overview of interpretability of machine learning," in *2018 IEEE 5th International Conference on data science and advanced analytics (DSAA)*, pp. 80–89, IEEE, 2018.
[36] S. Mohseni, N. Zarei, and E. D. Ragan, "A survey of evaluation methods and measures for interpretable machine learning," *arXiv preprint arXiv:1811.11839*, vol. 1, 2018.
[37] S. Mohseni, N. Zarei, and E. D. Ragan, "A multidisciplinary survey and framework for design and evaluation of explainable ai systems," *ACM Transactions on Interactive Intelligent Systems (TiiS)*, vol. 11, no. 3-4, pp. 1–45, 2021.
[38] D. V. Carvalho, E. M. Pereira, and J. S. Cardoso, "Machine learning interpretability: A survey on methods and metrics," *Electronics*, vol. 8, no. 8, p. 832, 2019.

Detecção de Anomalias em Usina Solar Fotovoltaica Conectada à Rede Elétrica no Brasil

Michelle Melo Cavalcante
Instituto Federal de São Paulo (IFSP)
Campinas, Brasil
michellemelo.c@gmail.com

João Lucas de Souza Silva
Instituto Federal de São Paulo (IFSP)
Campinas, Brasil
https://orcid.org/0000-0003-3206-2241

Samuel Botter Martins
Instituto Federal de São Paulo (IFSP)
Campinas, Brasil
samuel.martins@ifsp.edu.br

***Abstract*—O número de sistemas fotovoltaicos (FV) cresceu bastante no Brasil na última década, fato propiciado pelas resoluções e projetos de leis que foram implementados no decorrer dos anos, aliado a redução de custo da tecnologia. Paralelo a isso, surgiu o desafio de monitorar e garantir o cumprimento do retorno de investimento. Esse desafio existe devido as diversas anomalias presentes em tecnologias FV, desde os módulos FV até os equipamentos de processamento da energia. Tais anomalias podem ser detectadas/classificada por algoritmos de machine learning (ML) que são utilizados em outras aplicações. Assim, o presente artigo com base em dados reais de uma Usina FV, aplicou algoritmos de ML para testar sua capacidade nesses cenários. Para isso, foram testados os métodos (i) Isolation Forest, (ii) One-class SVM, e (iii) Amplitude interquartil. Como resultado verificou-se que o modelo utilizando One-Class SVM obteve melhor desempenho na detecção de anomalias, 26% inferir à amplitude interquartil na detecção de normalidades. Como conclusão, percebe-se que os modelos possuem potencial para serem aperfeiçoadas considerando a possibilidade de maior histórico de dados das medições da usina e de dados meteorológicos.**

***Keywords*—Usina Solar, Monitoração, Detecção de anomalias.**

I. Introdução

A implementação dos sistemas fotovoltaicos (FV) no Brasil acelerou no ano de 2022 com a implementação do Projeto de Lei nº 14.300 que regulamentou a Geração Distribuída (GD) no Brasil [1]. A nova lei fez com que investidores acelerassem a implantação dos sistemas FV para evitar pagamento de taxas a partir do ano de 2023, e irá garantir segurança juridica para os próximos anos, com viabilidade técnica e economica para os sistemas FV.

A GD é composta por fontes de geração que convertem energia próxima aos centros de cargas, dessa forma, reduzindo custos com transmissão de energia. No Brasil, a GD vem crescendo exponencialmente nos últimos anos [2]. Porém, com o crescimento surgem desafios como monitorar os sistemas FV para manter o tempo de retorno do investimento como planejado. Nesse caso, busca-se prevenir/detectar possíveis anomalias. Anomalias em sistemas FV podem ocorrer em toda cadeia de geração de energia, os mais comuns são nos módulos FV, inversores FV, e quadros de proteção.

Para os módulos FV pode-se dividir em falhas/anomalias irreversíveis e temporárias, ambas levam a redução da potência de trabalho do sistema [3]. As falhas irreversíveis são quando os módulos FV sofrem algum dano físico seja por problemas mecânicos, elétricos, ou até mesmo do ambiente, como uma chuva de granizo, necessitando nesses casos a substituição do módulo danificado. São diversos os problemas como corrosão, encapsulamento danificado/perda da transparência, condutores quebrados ou solda fria, rachaduras e queimaduras nas células [4]. Quanto as temporárias são geralmente causadas por sombreamento ou sujeira.

No caso dos inversores FV também pode-se dividir em irreversíveis e temporárias. As irreversíveis são geralmente problemas na eletrônica de potência que podem acontecer por oscilações da rede; temperatura; problemas na construção; resistências, indutâncias e capacitâncias parasitas; sobredimensionamento ou subdimensionamento; interferências eletromagnéticas, perdas por chaveamento, entre outros [5]-[7].

Já as temporárias podem ser causadas por temperatura, onde o inversor realiza o corte na potência de trabalho para reduzir a temperatura [8]; quedas de conexão devido distúrbios na rede elétrica; ou produção de harmônicas e outros distúrbios na rede elétrica [9].

Quanto as falhas nos quadros de proteção, são falhas que acabam desativando boa parte do sistema FV, já que são resultados de mau dimensionamento, ou seja, especificação de componentes errados, ou instalação incorreta, fato que acontece muito, dianta da não qualificação da mão-de-obra.

Dessa forma, com o desafio de prevenir/detectar falhas em Usinas FV, percebeu-se que a ciência de dados pode ser um aliado nesse cenário. A utilização de algoritmos de Machine Learning (ML), como (i) Isolation Forest, (ii) One-class SVM, ou (iii) Amplitude interquartil, podem ser interessante para detectar possíveis anomalias e ajudar o projetista na tomada de decisão. E tudo isso, sem a utilização de equipamentos de alto custo, como, por exemplo, o traçador de curva I-V que detectaria anomalias, porém, tem que parar a geração de energia da usina para medições.

Assim, o presente trabalho realiza a aplicação e estudo de algoritmos de ML para detecção de anomalias em Usinas FV. Para isso, foi utilizado um conjunto de dados reais de um sistema FV de 336,96 kWp [10]. Como contribuição científica é apresentado a aplicação dos métodos e metodologia utilizada. No artigo é tratado modelos que não

utilizam multiclasses, assim, detectando a anomalia, porém não classificando.

II. Trabalhos Relacionados

Na literatura existem vários trabalhos sobre monitoramento de Usinas FV, e aplicações de técnicas para detecção de anomalias. Por exemplo, em [11] foi realizado um *review* sobre monitoramento de Usinas FV. Os autores destacam que inspecionar periodicamente é importante para garantir a longevidade e desempenho do sistema. Porém, sabe-se que isso tem um determinado custo, e se for monitorar manualmente existe uma dificuldade sabendo-se que integradores FV, tem vários sistemas instalados de clientes.

Em [12] os autores utilizaram o Labview, que é um software pago para realizar o monitoramento das Usinas FV. No [13] os autores propuseram um sistema simples e de baixo custo com internet das coisas (IoT), para tanto, foi utilizado um ESPCAM 32. Semelhantemente, em [14], também é utilizado um sistema de IoT, entretanto, com o *MQTT broker*. Apesar desses sistemas serem capazes de monitorar e até mesmo enviar avisos no caso de desligamento da Usina, é necessário o suporte manual para detecção de análise de anomalias mais complexas como casos de sombreamento.

Outra forma de detectar anomalias explorada na literatura é o uso de imagens dos módulos FV. Em [15] é utilizado Aprendizagem Contrastiva Supervisionada aplicada em imagens de módulos FV em infravermelho. Inteligência artificial foi aplicada em outro caso com imagens termográficas em [16]. E, em [17] contém um conjunto de 36.543 imagens de testes de eletroluminescência para estudo de algoritmos capazes de detectar e distinguir os tipos de anomalias. O ponto negativo desse tipo de detecção de anomalias é a dificuldade e custo para obter as imagens, além disso, os defeitos em inversores FV ou quadros de proteção não são captados.

Semelhante ao presente artigo, tem-se aplicações de algoritmos de ML em dados obtidos através do monitoramento do próprio inversor FV ou sistema de medição de potência adicional. O ponto positivo dessa aplicação é o custo, e facilidade de acesso aos dados. Como ponto negativo, caso o monitoramento falhe, não é possível realizar a análise. Outro ponto é a dificuldade de identificar a origem de alguns problemas. Existirão casos em que uma análise em campo será necessária.

Como destaque, tem-se o artigo [18] em que os autores estudaram alguns tipos de aplicações de ML, como *AutoEncoder Long Short-Term Memory (AE-LSTM), Facebook-Prophet, e Isolation Forest,* bem como, citam outros trabalhos semelhantes. Os resultados foram satisfatórios, e os autores destacaram que é preciso também investigar e aplicar técnicas inteligentes de mitigação de anomalias. Em [19] os autores aplicaram ML para classificar possíveis anomalias (Fig. 1) em um pequeno sistema FV com sucesso. E, por fim, [20] elaborou um compilado das possíveis aplicações de ML em Usinas FV.

III. Algoritmos de ML

Diferentes técnicas e métodos foram utilizados neste artigo e são discutidos nesta seção. Os algoritmos de ML usados são (i) Isolation Forest, (ii) One-class SVM e, utilizando um viés mais estatítico, a (iii) Amplitude interquartil. Essas arquiteturas de algoritmos são discutidas para criar uma compreensão da metodologia de pesquisa.

A. Isolation Forest

Isolation Forest é um modelo de detecção de anomalias não supervisionado construído em árvores de decisão. O método define anomalias como pontos de dados que são limitados e anormais [21]. A partir dessa definição e do entedimento de que como os outliers são a minoria e se destoam em um conjunto de dados, estes são mais suscetíveis a serem isolados, ou seja, são mais suscetíveis a serem separados dos demais pontos [22].

Para isso, o Isolation Forest busca particionar um determinado conjunto de dados de maneira aleatória, selecionando aleatoriamente um atributo e por fim, definindo um valor também aleatório da proporção da divisão a ser feita sobre o conjunto de dados. Este valor, no entanto, irá variar entre o valor máximo e mínimo do atributo selecionado. Esse processo acontece recursivamente, até que todos os pontos consigam ser isolados [22].

O algoritmo pode ser descrito da seguinte forma:

$$s(p, n) = 2\hat{}(- E(h(p))/c(n)), \quad (1)$$

onde $E(h(p))$ é o caminho médio até o ponto p entre uma coleção de árvores e $(c(n))$ é o caminho médio até p dadas n instâncias. Pode-se calcular $(c(n))$ por:

$$c(n) = 2H(n - 1) - (2(n - 1)/n), \quad (2)$$

onde H é o número harmônico e pode ser estimado por ln(n) = 0.5772156649. Dessa forma, é possível afirmar que um ponto p será considerado ou não uma anomalia se:

- Se s for um valor próximo a 1, então é provável que este ponto seja um outlier;
- Se s for um valor muito menor que 0.5, então é possível afirmar com certa segurança que o ponto é um inlier;
- Se s for um valor próximo de 0.5 ($s \approx 0.5$), então toda a amostra não possui nenhuma anomalia distinta.

B. One-class SVM

O One-Class SVM foi desenvolvido por Arnold [23] e foi modificado para computar as entradas do kernel dinamicamente devido a limitações de memória.

O algoritmo One-Class SVM funciona mapeando os dados de entrada em um espaço de recursos de alta dimensão (através de um kernel) e encontra iterativamente o hiperplano de margem máxima que melhor separa os dados de treinamento da origem.

O One-Class SVM pode ser visto como um SVM regular de duas classes, onde todos os dados de treinamento estão na primeira classe e a origem é tomada como o único membro da segunda classe. Assim, o hiperplano (ou limite de decisão linear) corresponde à regra de classificação:

$$f(x) = w \cdot x + b, \quad (3)$$

onde w é o vetor normal e b é o bias. O One-Class SVM resolve um problema de otimização para encontrar a regra com margem geométrica máxima. Podemos usar essa regra de classificação para atribuir um rótulo a um exemplo de teste x. Se rotularmos $f(x) < 0$ como uma anomalia, o caso contrário é rotulado como normal.

Na prática, existe um compromisso entre maximizar a distância do hiperplano da origem e o número de pontos de dados de treinamento contidos na região separada da origem pelo hiperplano.

C. *Amplitude Interquartil*

Diferente das modelagens utilizando inteligência artificial com o Isolation Forest e o One-Calss SVM, avaliou-se também os outliers. Considerou-se outlier toda e qualquer potência de saída do inversor fora da amplitude interquatílica na proporção de 1,5*IQR. Um exemplo, é o resultado apresentado na Fig. 1.

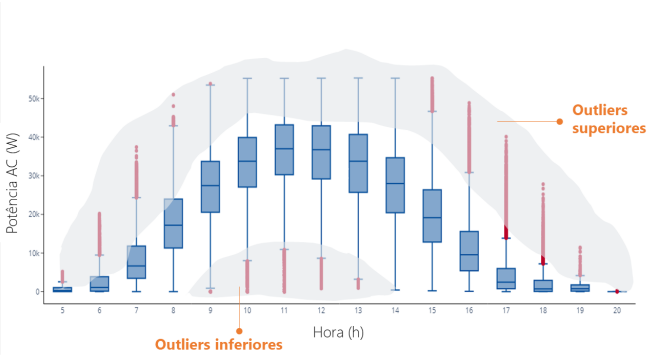


Fig. 1. Agrupamento das potências e horas para identificação dos outliers.

IV. Estudo de Caso

A pesquisa usou como estudo de caso uma das instalações do projeto Campus Sustentável da Universidade de Campinas (UNICAMP) localizada no Campus de Campinas, em São Paulo (Brasil).

A. *Usina Fotovoltaica*

A Usina fotovoltaica (Fig. 2) está localizada no ginásio da UNICAMP e conta com uma potência total de 336,8 kWp e geração estimada de 481.160 kWh por ano.

Fig. 2. Usina Fotovoltaica instalada na UNICAMP.

B. *Dados utilizados*

Ao total obteve-se 1 ano de registro com intervalos de 15 minutos. As informações registradas foram tensões, correntes, frequência da rede, potência ativa e reativa de saída dos inversores, fator de potência e Energia gerada.

Para simplificar, utilizou-se os dados de potência de saída dos inversores que corresponde a relação da tensão e corrente convertidas pelos módulos FV. Com os dados de potência de saída do inversor, é possível identificar problemas em toda a cadeia que antecede a geração da energia: problemas nos módulos, na rede elétrica ou no próprio inversor.

C. *Anomalias identificadas durante a exploração dos dados*

Ao plotar o período de 20 de julho de 2021 e 1 de agosto de 2021 (conforme Fig. 3), foram identificados 413 dados anômalos e 2407 dados normais. As anomalias foram:

- Falta: tensão vai a zero por problema na rede identificado pela frequência nula
- Queda de tensão provocada pelo clima. Provavelmente um dia chuvoso.
- Oscilações causadas por nuvens ou sombreamento.

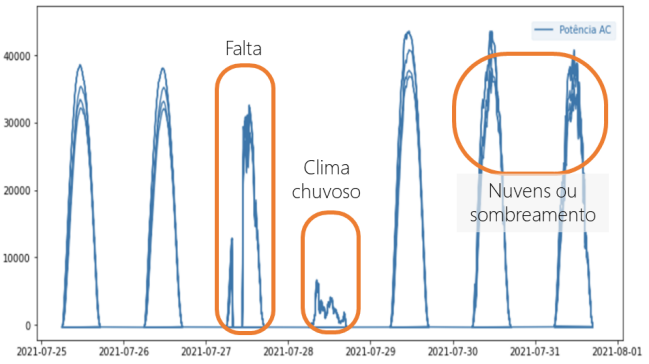


Fig. 3. Anomalias identificadas durante fase de exploração dos dados.

V. Detecção de Anomalias em Usina Solar Fotovoltaica Conectada à Rede Elétrica

Esta seção discute a avaliação do estudo de caso realizado para validar e avaliar as reivindicações, relatando as descobertas e resultados em detalhes. Conforme relatado na seção III, utilizou-se (i) Isolation Forest, (ii) One-class SVM e, utilizando um viés mais estatítico, a (iii) Amplitude interquartil.

A. Isolation Forest

Isolation Forest detecta anomalias puramente com base no fato de que anomalias são pontos de dados que são poucos e diferentes. O isolamento das anomalias é implementado sem empregar nenhuma medida de distância ou densidade. Este método é fundamentalmente diferente dos algoritmos baseados em agrupamento ou distância.

Foram identificadas aproximadamente 17% das anomalias verdadeiras e 85% dos dados considerados normais.

B. One-Class SVM

Para o One-Class SVM os principais parâmetros utilizados foram:

- O limite superior na fração de erros de treinamento e um limite inferior na fração de vetores de suporte (nu) igual a 0,01
- tipo de kernel (kernel) padrão: "rbf"
- Coeficiente do kernel (gamma) igual a 0,01.

Foram identificadas aproximadamente 30% das anomalias verdadeiras e 71% dos dados considerados normais.

C. Amplitude Interquartil

Na amplitude interquartil utilizou-se os outliers. Confome mencionado anteriormente, considerou-se outlier toda e qualquer potência de saída do inversor fora da amplitude interquatílica (boxplot) representado pela Fig. 1.

D. Resumo dos resultados obtidos

Considerou-se o uso da acurácia balanceada devido ao desbalanceamento da base condizente com aspectos de anomalias que ocorrem com bem menos frequência quando comparada aos valores normais medidos.

TABLE I. Comparação entre os métodso de Isolation Forest, One-Class SVM e Amplitude Interquartil

		Métricas			
		Precisão	***Recall***	***F1-Score***	***Acurácia***
Isolation Forset	Anomalia	0,18	0,18	0,18	0,52
	Normal	0,87	0,87	0,87	
One-class SVM	Anomalia	0,16	0,34	0,22	0,53
	Normal	0,88	0,73	0,79	
Amplitude interquartil	Anomalia	0,83	0,25	0,39	0,62
	Normal	0,90	0,99	0,94	

Os resultados gráficos e da matriz de confusão são apresentados na sequência através das Fig. 4 e 5, respectivamente.

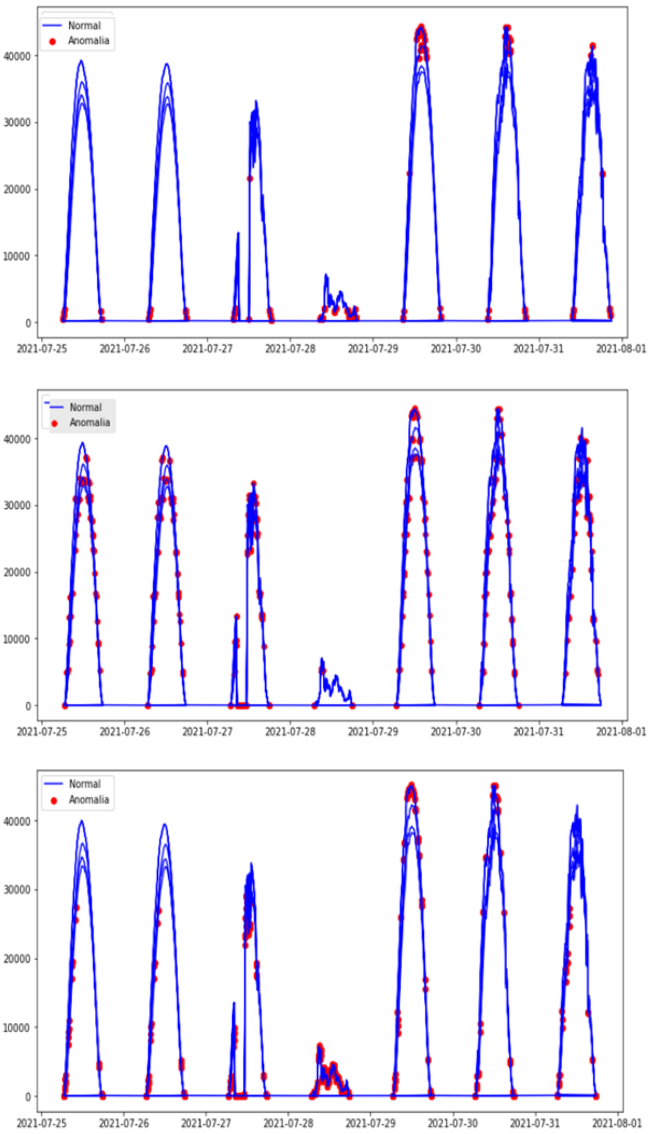


Fig. 4. Anomalias identificadas ao usar Isolation Forest, One-Class SVM e Amplitude Interquartil, respectivamente.

Nota-se que o modelo One-Class SVM foi mais eficiente para detectar anomalias (verdadeiro positivo), porém foi a que menos acertou a normalidade. De maneira geral, os modelos para detectar as anomalias utilizando o limite fora do Intervalo Interquartil e Isolation Forest demonstraram melhor eficiência na classificação da normalidade (verdadeiro negativo).

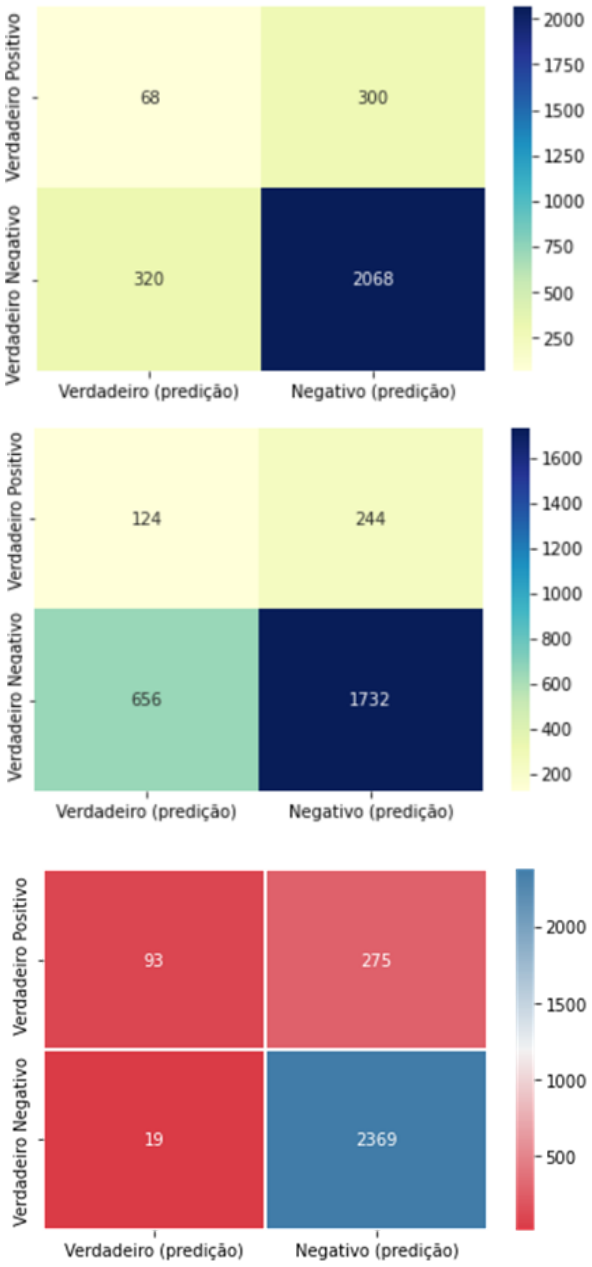


Fig. 5. Matriz confusão para Isolation Forest, One-Class SVM e Amplitude Interquartil, respectivamente.

Conclusão

A proposta mostrou-se promissora para ser evoluída em situações em que não há parâmetros ambientais, ou seja, monitorar plantas de instalações fotovoltaicas não equipadas com estação meteorológica, com custo baixo, sem adicionar novos equipamentos. Além disso, a obtenção de um histórico maior de medições de potência e geração da usina podem ajudar no treinamento e modelos mais assertivos.

Como trabalhos futuros, pretende-se aperfeiçoar a detecção das anomalias para obter maior exatidão e incluir, a classificação das anomalias (anomalias causadas por chuva, neve, clima nublado, problemas na tensão da rede, entre outros).

Agradecimentos

Os autores do artigo agradecem e destacam a importância do Projeto Campus Sustentável da UNICAMP (P&D e PEE - ANEEL e CPFL) que gerou um conjunto de trabalhos e livro, que incentivam e fornecem meios para idealização desse e outros trabalhos. Além disso, agradecem ao Instituto Federal de Educação, Ciência e Tecnologia de São Paulo, sobretudo, ao Programa de Pós-Graduação Lato Sensu em Ciência de Dados.

Referências

[1] Brasil. LEI Nº 14.300, DE 6 DE JANEIRO DE 2022. Disponível em: https://www.in.gov.br/en/web/dou/-/lei-n-14.300-de-6-de-janeiro-de-2022-372467821. Acesso em: 07 de jan. de 2023.

[2] SILVA, J. L. S.; CAVALCANTE, M. M. ; MACHADO, R. ; RIBEIRO, M. S. ; DELGADO, D. B. M. ; CARVALHO, M. . Análise do crescimento da geração distribuída: Estudo de caso do Brasil com ênfase no estado de Minas Gerais. REVISTA DE ENGENHARIA E TECNOLOGIA, v. 10, p. 169-183, 2018.

[3] Urbanetz, I. V. (2019). Diagnóstico de falhas em módulos fotovoltaicos. Mestrado em Energias Renováveis e Eficiência Energética. Escola Superior de Tecnologia e Gestão. Bragança, Portugal.

[4] D. Sera e R. Teodorescu, "Robust series resistance estimation for diagnostics of photovoltaic modules," Industrial Electronics, 2009. IECON'09. 35th Annual Conference of IEEE, p. 800–805, 2009.

[5] LUJARA, N. K.; WYK, J. D. V.; MATERU, P. N. Power Electronic Loss Models of dc-dc Converters in Photovoltaic Applications. IEEE International „Symposium on Industrial Electronics. Proceedings. ISIE'98, p. 1Ú6, 1998

[6] CHAN, P.-W.; MASRI, S. DC-DC Boost Converter with Constant Output Voltage for Grid Connected Photovoltaic Application System. International Conference on Intelligent and Advanced Systems, v. 21, n. 4, p. 67Ú99, 2010.

[7] TANG, L.; SHI, Z.; YANG, X. Ventilation Analysis and Simulation for Inverter of Photovoltaic Power Plant. Procedia Engineering. Elsevier B.V., v. 205, p. 1820Ú1827, 2017. ISSN 18777058. Disponível em: .

[8] J. L. de Souza Silva, K. B. de Melo, T. S. Costa, G. M. Vieira Machado, H. S. Moreira and M. G. Villalva, "Impact of Bifacial Modules on the Inverter Clipping in Distributed Generation Photovoltaic Systems in Brazil," 2021 Brazilian Power Electronics Conference (COBEP), João Pessoa, Brazil, 2021, pp. 1-6, doi: 10.1109/COBEP53665.2021.9684055.

[9] ELTAWIL, M. A.; ZHAO, Z. Grid-connected photovoltaic power systems: Technical and potential problems-A review. Renewable and Sustainable Energy Reviews, v. 14, n. 1, p. 112Ú129, 2010. ISSN 13640321.

[10] DE SOUZA SILVA, JOÃO LUCAS; BARBOSA DE MELO, KAREN ; DOS SANTOS, KAIO VIEIRA ; YOITI SAKO, ELSON ; KITAYAMA DA SILVA, MICHELLE ; SOEIRO MOREIRA, HUGO ; BOLOGNESI ARCHILLI, GIULIANNO ; ITO CYPRIANO, JOÃO GUILHERME ; CAMPOS, RAFAEL ESPINO ; PEREIRA DA SILVA, LUIZ CARLOS ; GRADELLA VILLALVA, MARCELO . Case study of photovoltaic power plants in a model of sustainable university in Brazil. RENEWABLE ENERGY, v. 196, p. 247-260, 2022.

[11] Ejgar, M., & Momin, B. (2018). Solar plant monitoring system: A review. Proceedings of the International Conference on Computing Methodologies and Communication, ICCMC 2017, 2018-January(Iccmc), 1142–1144. https://doi.org/10.1109/ICCMC.2017.8282652

[12] G. Bayrak and M. Cebeci, "Monitoring a grid connected pv power generation system with labview," in Renewable Energy Research and Applications (ICRERA), 2013 International Conference on. IEEE, 2013, pp. 562–567.

[13] M. D. Bhujbal and M. G. Unde, "Real Time Monitoring and Security of Solar Power Plant Using IoT," 2022 IEEE India Council International Subsections Conference (INDISCON), Bhubaneswar, India, 2022, pp. 1-5, doi: 10.1109/INDISCON54605.2022.9862817.

[14] J. M. Ramadhan, R. Mardiati and I. N. Haq, "IoT Monitoring System for Solar Power Plant Based on MQTT Publisher / Subscriber Protocol," 2021 7th International Conference on Wireless and Telematics (ICWT), Bandung, Indonesia, 2021, pp. 1-6, doi: 10.1109/ICWT52862.2021.9678503.

[15] Bommes, L., Hoffmann, M., Buerhop-Lutz, C., Pickel, T., Hauch, J., Brabec, C., Maier, A., & Marius Peters, I. (2022). Anomaly detection in IR images of PV modules using supervised contrastive learning. Progress in Photovoltaics: Research and Applications, 30(6), 597–614. https://doi.org/10.1002/pip.3518

[16] G. Cipriani, D. Manno, V. D. Dio and M. Traverso, "Thermal anomalies detection in a photovoltaic plant using artificial intelligence: Italy case studies," 2021 IEEE International Conference on Environment and Electrical Engineering and 2021 IEEE Industrial and Commercial Power Systems Europe (EEEIC / I&CPS Europe), Bari, Italy, 2021, pp. 1-6, doi: 10.1109/EEEIC/ICPSEurope51590.2021.9584494.

[17] B. Su, Z. Zhou and H. Chen, "PVEL-AD: A Large-Scale Open-World Dataset for Photovoltaic Cell Anomaly Detection," in IEEE Transactions on Industrial Informatics, vol. 19, no. 1, pp. 404-413, Jan. 2023, doi: 10.1109/TII.2022.3162846.

[18] Ibrahim, M., Alsheikh, A., Awaysheh, F. M., & Alshehri, M. D. (2022). Machine Learning Schemes for Anomaly Detection in Solar Power Plants. Energies, 15(3), 1–17. https://doi.org/10.3390/en15031082

[19] T. Babasaki and Y. Higuchi, "Using PV string data to diagnose failure of solar panels in a solar power plant," 2018 IEEE International Telecommunications Energy Conference (INTELEC), Turino, Italy, 2018, pp. 1-4, doi: 10.1109/INTLEC.2018.8612400.

[20] B. Khelifi, M. A. Zdiri and F. Ben Salem, "Machine Learning for Solar Power Systems-A short tour," 2021 12th International Renewable Energy Congress (IREC), Hammamet, Tunisia, 2021, pp. 1-6, doi: 10.1109/IREC52758.2021.9624896.

[21] Hariri, S.; Kind, M.C.; Brunner, R.J. Extended isolation forest. IEEE Trans. Knowl. Data Eng. 2019, 33, 1479–1489.

[22] Guimarães, Gabriel Bueno. "Uma análise exploratória da influência da detecção de outliers na precificação de produtos em e-commerce.". Universidade Federal de Ouro Preto, 2022.

[23] A. Arnold. SVM anomaly detection code. IDS Lab, Columbia University, 2002.

Impacto do pré-processamento em análise de sentimentos utilizando PLN

Daniel Vargas Shimamoto
Instituto Federal de Educação, Ciência e Tecnologia de São Paulo - Campus:Campinas.
Campinas, Brasil
daniel.shimamoto@aluno.ifsp.edu.br

Ricardo Barz Sovat
Instituto Federal de Educação, Ciência e Tecnologia de São Paulo - Campus:Campinas.
Campinas, Brasil
sovat@ifsp.edu.br

***Resumo* — A análise de sentimentos é um dos problemas clássicos que envolvem o processamento de linguagem natural. Esse tipo de abordagem vem sendo estudada ao longo dos anos e diversas técnicas foram sendo desenvolvidas para buscar o melhor formato de implementá-lo. Um dos desafios relacionado a esse tipo de problema é a barreira linguística: cada idioma possui uma particularidade e a transposição de técnicas de um formato de linguagem para outro nem sempre apresenta o mesmo comportamento. Além disso, existem diversas técnicas que podem ser empregadas para tratar, representar e classificar os dados. Baseado nessas questões, este trabalho analisa o impacto do pré-processamento dos dados em diversos formatos de classificação, combinando diferentes técnicas de vetorização, número de vetores e classificadores. Esta análise inclui três conjuntos de dados obtidos de *marketplaces* brasileiros com comentários em português, unificados e separados em cinco diferentes bases para garantir reprodutibilidade. Foram testados sete tipos de pré-processamentos diferentes em vinte formatos diferentes *pipelines* de classificação, possibilitando comparar os efeitos de cada um dos processos nas métricas avaliadas.**

***Palavras-chave* —** *PLN, Análise de sentimentos, pré-processamento, vetorização, aprendizado de máquina.*

I. Introdução

O processamento de linguagem natural (PLN) é um campo de estudo que busca desenvolver sistemas computacionais para processar e compreender as linguagens humanas. Essa área é uma ramificação da computação que envolve conhecimentos de inteligência artificial e linguística [1]. A aplicação da PLN é ampla, sites de buscas a utilizam por meio do preenchimento automático e uso de sinônimos, assim como os serviços de e-mails com os filtros de spam. Um uso mais específico é a análise de sentimentos: esta aplicação busca entender e monitorar as sensações de um público em resposta a um estímulo, seja este uma mudança, uma notícia ou uma ação [2].

A análise de sentimentos vai ao encontro da transformação *customer centric* que as empresas estão buscando. Por meio da filosofia do "cliente primeiro", as grandes companhias estão em busca de entender as necessidades dos clientes e se aproximar deles, explorando o desenvolvimento das novas tecnologias de computação em nuvem, redes sociais e dados [3]. A ideia por trás disso é simples: os clientes não estão mais em busca de produtos ou serviços simplesmente, eles buscam contratar experiências que satisfaçam suas necessidades, vontades e desejos. Dessa forma, as análises mais tradicionais de mapear os padrões de compras e fornecer indicações de produtos relacionados, que foi um grande diferencial no início da análise de dados de consumidores, já não trazem uma proposta de valor para os clientes em si só. Assim, é preciso dar um passo adiante e entender não só os efeitos que as decisões das empresas têm no padrão de consumo dos clientes, mas quais as percepções que eles têm sobre isso, sendo um espaço para a aplicação de análise de sentimentos [4].

Diferente da linguagem de programação tradicional, onde o computador recebe uma série de comandos definidos que devem ser executados por meio de um conjunto finito de operações [2], o sistema de comunicação humana envolve uma combinação complexa de componentes, como palavras, sentenças e contextos, que tornam a interpretação da informação mais difícil para a máquina [1]. Um desafio para um problema que envolva PLN é transformar os dados textuais em dados numéricos para que o computador entenda e consiga processar essas informações [2]. Outro importante fator para PLN é a barreira linguística: os pré-processamentos utilizados na base de dados devem considerar as particularidades e as características de cada idioma, aumentando a complexidade dos problemas.

Um processo de análise de sentimentos utilizando PLN possui três etapas principais. A primeira é a limpeza e pré-processamento, em que os dados são submetidos a uma série de tratamentos a fim de deixar a base a mais coesa possível [5]. É nesta etapa que as informações textuais passam por processos de normalização, retirada de informações que podem afetar a análise, como *stop words* e pontuações e redução das palavras (*Stemming* ou *Lemmatization*) [6].

Após essa etapa é necessário adequar o formato dos dados de modo que o computador consiga interpretá-lo: a vetorização. O objetivo desta etapa é criar uma representação matemática, seja utilizando números ou vetores, de uma unidade linguística. Existem diversas técnicas e estratégias que podem ser utilizadas para realizar essas operações [1]. A última etapa consiste em aplicar um classificador baseado em técnicas de aprendizado de máquinas para agrupar e categorizar os dados [7].

As técnicas de pré-processamento, vetorização e classificação foram sendo desenvolvidas com o passar dos anos, novos modos e métodos mais rebuscados foram desenvolvidas nas três etapas. O objetivo deste trabalho é comparar a influência dessas etapas na análise binária de sentimentos em uma base de comentários sobre aquisição de produtos em marketplaces brasileiros.

II. Experimento

O estudo experimental conduzido nesta seção é composto das seguintes fases: explicação das bases de dados, preparação dos dados, pré-processamento, descrição

dos tipos de vetorização, descrição dos modelos de classificação, métricas de avaliação usadas e teste de hipótese.

A. Base de dados

Foram utilizadas as bases de dados "Brazilian Portuguese Sentiment Analysis Datasets" [8], onde são compilados dados em português de diferentes origens. Para essa análise são utilizadas as seguintes bases:

Olist: Este conjunto de dados contém mais de 100 mil pedidos realizados entre os anos de 2016 e 2018 pela plataforma Olist, a maior loja de departamentos dos marketplaces brasileiros, conectando as pequenas empresas do Brasil com os consumidores em um único canal [9].

B2W: O *corpus* B2W-Reviews01 é uma base de dados de análises de produtos contendo mais de 130 mil avaliações de clientes entre janeiro e maio de 2018 coletadas do site Americanas.com [10].

Buscapé: O *corpus* Buscapé é um conjunto de dados retirados do site de mesmo nome, uma página de busca de produtos e preços, que contém mais de 80 mil avaliações retiradas em Setembro de 2013 [11].

Os dados das três bases contêm duas informações principais: o comentário e a nota de avaliação. As notas variam de 1 a 5 para os dados da Olist e B2W e de 0 a 5 para a base do Buscapé. A fim de equiparar as bases, os comentários atribuídos à avaliação 0 desta foram desconsiderados. Para rotular os dados a fim de transformar as bases em um estudo relacionado à análise de sentimentos binária, as notas de classificação foram separadas em 4 e 5 como comentários positivos, 1 e 2 como negativo, desconsiderando os comentários com 3 estrelas [8].

B. Preparação dos dados

Com os dados devidamente rotulados, foram removidas as informações em que a polaridade era nula. A fim de equiparar as bases e balancear as informações, retirou-se uma amostra tomando como base a polaridade com menor número de comentários, conforme tabela 1.

Tabela 1. Número de amostras selecionado de cada dataset

Dataset	***Comentários***	***Polaridade nula***	***Polaridade Negativa***	***Polaridade Positiva***
Olist	41.744	16.315	11.408	26.671
B2W	132.373	3.665	35.758	80.300
Buscapé	84.991	11.365	6.810	66.816
Total	259.108	31.345	53.976	173.787

As amostras selecionadas passaram pelas etapas de pré-processamentos que serão descritas no próximo tópico e delas foram excluídos os dados que geraram algum comentário com o valores nulos após os tratamentos. Após essa etapa as bases foram concatenadas a fim de gerar um dataset único. Não existe um número mínimo específico indicado para definir o tamanho de um dataset para treinar um modelo de análise de sentimentos, mas alguns estudos apontam que dependendo da composição do vocabulário [12], dados a partir de 10 mil amostras já são adequados para garantir uma análise [13].

Com base nessas informações e buscando garantir maior reprodutibilidade dos resultados, as bases foram unificadas e divididas em 5 partes, contendo 20 mil amostras com a mesma proporção de comentários positivos e negativos.

C. Pré-processamento

Os pré-processamentos foram divididos em três grupos:

Mínimo (*min*): Este tratamento é considerado o mais fundamental em um problema envolvendo PLN. Ele consiste em normalizar as palavras, escrevendo-as em letras minúsculas e evitando que os caracteres não alfabéticos atrapalhem o entendimento do modelo. Além disso, são removidas algumas informações que não são consideradas fundamentais para a análise, como pontuação e acentuação [5]. Complementando esses processos, os números foram padronizados para o numeral 0, seguindo o formato de pré-processamento utilizado pelo NILC [14]. Para facilitar a tokenização do texto, foram retirados também os códigos de quebra de linha (\n) e os espaços duplicados.

Stop words (*stop*): Algumas classes de palavras, como artigos e preposições, são muito comuns na linguagem, mas não acrescentam significado na frase. Essas palavras são conhecidas como *stop words* e acrescentam muito ruído aos dados, sendo uma boa prática removê-las da base de dados. A biblioteca do NLTK possui uma lista com as *stop words* mais comuns, com suporte para o português [2]. Neste pré-processamento, foram retiradas as *stop words* conforme lista padronizada oferecida pela biblioteca NLTK na versão 3.6.1.

Stemming (*ste*) ou stemização: Existem dois métodos principais que podem ser utilizados para encontrar similaridade entre palavras, a stemização é uma técnica utilizada para reduzir uma palavra ao seu radical, de modo que todas as variações das palavras possam ser representadas da mesma forma. Embora essa redução possa reduzir a palavra a uma classe gramatical incorreta, ela é muito usada na classificação de documentos, pois foca no sentido geral de um texto em vez do seu significado mais profundo. Um dos algoritmos mais usados para fazer essa redução é o Porter Stemming, método disponível na biblioteca NLTK através da classe PorterStemmer [6]. Uma forma mais avançada de redução de palavras é a *Lemmatization* (lematização). Nesse processo cada palavra é mapeada em todas as diferentes formas que ela pode ter, buscando encontrar a sua forma básica de representar, o seu lema. Embora sua definição seja próxima do processo de stemização, seu resultado é sempre uma palavra que existe na gramática. Um exemplo é a palavra "melhor" que se aplicado stemização seria mantida igual, mas na lematização é reduzida a "bom". Embora mais robusto, esse processo requer um maior conhecimento linguístico e o desenvolvimento de algoritmos eficientes é um problema em aberto na pesquisa de PLN até hoje[1]. Para este estudo foi utilizado o método RSLPStemmer da biblioteca NLTK na versão 3.6.1.

Para encontrar a sinergia entre os pré-processamentos, todas as etapas foram combinadas em pares e também foi realizada a união das três técnicas. As informações da ordem dos pré-processamentos podem ser vistas na tabela 2. A tabela 3 tem um exemplo de um comentário transformado quando submetido a cada um dos processos.

Tabela 2. Informações de pré-processamento

Tratamento	*Pré-processamento I*	*Pré-processamento II*	*Pré-processamento III*
min	Mínimo	-	-
stop	StopWords	-	-
ste	Stemização	-	-
min_stop	StopWords	Mínimo	-
min_ste	Mínimo	Stemização	-
stop_ste	StopWords	Stemização	-
min_stop_ste	StopWords	Mínimo	Stemização

Tabela 3. Exemplo de comentário submetido aos pré-processamento

Tratamento	*Comentário*
Original	O produto é inferior ao anunciado,baixa qualidade
min	o produto e inferior ao anunciado baixa qualidade
stop	produto inferior anunciado,baixa qualidade
ste	o produt é inferi ao anunciado, baix qual
min_stop	produto inferior anunciado baixa qualidade
min_ste	o produt e inferi ao anunci baix qual
stop_ste	produt inferi anunciado,baix qual
min_stop_ste	produt inferi anunci baix qual

D. Vetorização

TF-IDF: Esta técnica é baseada na ponderação dos termos mais comuns usados em um documento em relação aos demais. Considerando que, caso uma palavra apareça várias vezes em um documento e poucas nos demais ela deva ter uma importância maior no documento analisado, este método tem como foco identificar a relevância das palavras no texto [15]. O termo TF (frequência do termo) calcula a pontuação de frequência de uma palavra em um documento, enquanto o IDF (frequência inversa do documento) resulta no valor de raridade de uma palavra em todos os documentos. O resultado gera uma ponderação indicando que nem todas as palavras possuem grau de importância igual nos documentos, de modo que quanto maior o valor de TF-IDF para um determinado termo, maior a indicação que ele é uma palavra distinta e pode possuir uma informação útil [6]. Neste estudo foi utilizada a função TfidfVectorizer da biblioteca sklearn versão 1.0.1.

Word Embeddings (Word2vec): A abordagem *word embeddings* possui uma representação distribuída e busca representar uma palavra com base no contexto em que ela está inserida, levando em conta que as palavras mais próximas tendem a ter um efeito direto nas palavras ao seu redor [2]. Um dos algoritmos mais famosos é o Word2Vec que permite representar uma palavra com vetores densos de baixa dimensão (entre 50 e 500 vetores) [1]. Este modelo permite o treinamento em dois tipos de arquiteturas. O Continuos Bag-of-Words (CBOW) tenta prever a palavra com base no contextos que elas são inseridas, considerando as palavras ao seu redor. Já no Skip-gram, o objetivo é prever a janela de palavras ao redor da palavra de entrada [2]. A figura 1 apresenta os modelos de arquitetura propostos por Mikolov [16] . Foi utilizada a implementação Word2Vec da biblioteca gensim versão 4.2.0, em ambas arquiteturas, CBOW e Skip-gram, considerando uma janela de 3 palavras excluindo termos que tenham uma ocorrência menor que 5, para evitar que palavras pouco recorrentes ou que foram escritas incorretamente interfiram nos modelos.

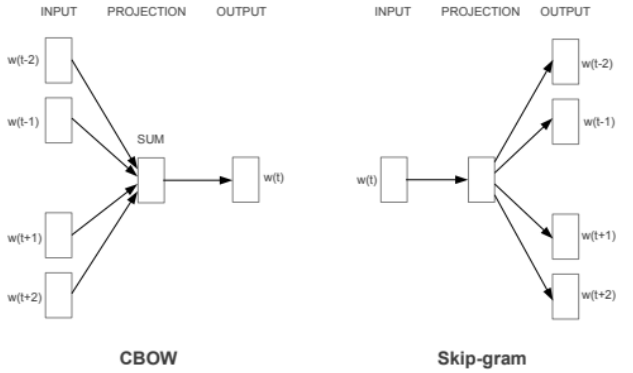


Fig 1. Modelo de arquitetura Word2Vec. A arquitetura CBOW prevê a palavra com base no contexto, enquanto o Skip-gram prediz as palavras ao redor da palavra foco. Figura extraída de [16].

NILC: O NILC-Embedding criado pelo Núcleo Interinstitucional de Linguística Computacional (NILC) é um repositório que tem como objetivo disponibilizar vetores de palavras prontos na Língua Portuguesa. Os vetores foram gerados de mais de 3,8 milhões de textos utilizando textos brasileiros e europeus. A base foi submetida em um pré-processamento simples, substituindo palavras com menos de cinco ocorrências pelo símbolo "UNKNOWN", numerais normalizados em 0, URL pelo token URL e e-mails pelo token EMAIL. Dentre os modelos treinados, foi utilizado o Word2Vec com as variações CBOW e Skip-gram [14], informações que foram importadas e aplicadas neste estudo.

Os modelos baseados em Word Embeddings geram vetores numéricos representando palavras e precisam gerar vetores de textos para representar os comentários. Dentre as estratégias possíveis, optou-se por somar os vetores de palavras para gerar um vetor textual resultante por conta da capacidade de processamento. Outro ponto importante é a definição do tamanho dos vetores, levando em conta os modelos pré-treinados do NILC, foram utilizados vetores de 100 e 300 dimensões para esse estudo. A base foi previamente separada em 80% para treino e 20% para teste, utilizando apenas os dados de treino para as funções de vetorização.

E. Modelos de classificação

Naive Bayes: O Classificador Naive Bayes (NB) é muito utilizado na análise de sentimentos. O modelo utiliza um conjunto de dados para calcular a probabilidade do vetor de entrada pertencer a cada um dos possíveis rótulos, inferindo a categoria com maior probabilidade a esse vetor [17]. O NB utiliza como base o Teorema de Bayes, que diz ser possível calcular a probabilidade de uma hipótese dada uma evidência, usando a probabilidade da evidência dada a hipótese e as probabilidades incondicionais da hipótese e da evidência [18]. A equação (1) ilustra o Teorema de Bayes.

$$p\,(H|E) = \frac{p(E|H) * p(H)}{p(E)} \quad (1)$$

Na classificação de textos, o Teorema de Bayes calcula a probabilidade de uma frase pertencer a uma determinada classe (p(H|E)). Para isso é utilizada a probabilidade da classe conter a frase, ou seja, dentro da classe analisada, qual é a probabilidade de uma das frases ser igual à frase investigada (p(E|H)). Além disso, são computadas a probabilidade da classe aparecer no treinamento (p(H)) e a probabilidade da frase aparecer nos dados de treino (p(E)) [19]. O modelo implementado foi uma variação do Naive Bayes, o GaussianNB (GNB) da biblioteca sklearn versão 1.0.1.

Regressão Logística: A regressão logística é um modelo muito utilizado em classificações binárias. Ele consiste em calcular a soma ponderada das variáveis de entrada adicionando um viés para gerar uma saída indicando a probabilidade do resultado pertencer a classe positiva ou negativa. Esse modelo possui alguns hiperparâmetros que podem ser otimizados, entre eles os relacionados ao tipo e ao peso da regularização aplicada [7]. Foi utilizado o *LogisticRegression* da biblioteca sklearn versão 1.0.1.

Para garantir maior confiabilidade nos resultados, os modelos foram treinados utilizando o GridSearchCV com 5 *folds*. No modelo de Regressão Logística foram otimizados os hiperparâmetros de *penality* (l1, l2) e *C* (0.01, 0.1, 1), além de utilizar o *solver* liblinear e alterado o *max_iter* para 400.

F. Métricas de avaliação

Acurácia (*Accuracy*): A acurácia é uma medida que sumariza o erro total de uma amostra. Ela verifica o número de classificações corretas do modelo, considerando as duas classes [20]. É calculada segundo a equação (2).

$$Accuracy = \frac{\sum True\ Positive + \sum TrueNegative}{SampleSize} \quad (2)$$

Precisão (*Precision*): A precisão é uma métrica que verifica a acurácia das predições positivas, ou seja, das classificações positivas, quantas de fatos foram previstas como positivas [7]. Pode ser representada pela equação (3).

$$Precision = \frac{\sum TruePositive}{\sum TruePositive + \sum FalsePositive} \quad (3)$$

Recall: O *recall* é a proporção de instâncias positivas que são detectadas corretamente pelo modelo de classificação. Diferente da precisão, o recall mensura dentre todas as observações classificadas pelo modelo como positiva, quantas ele acertou [7]. A equação (4) representa o recall.

$$Recall = \frac{\sum TruePositive}{\sum TruePositive + \sum FalseNegative} \quad (4)$$

O uso dessas três métricas ajuda a entender se existe algum viés no modelo, em que ele possa estar classificando melhor uma classe do que a outra, além de avaliar o seu desempenho. Para este estudo, foram computadas as métricas de acurácia para os dados de treino e teste e precisão e recall apenas os dados de teste.

G. Teste de hipóteses

Qui Quadrado: O teste qui quadrado foi utilizado para verificar se as métricas de avaliação possuem uma diferença estatística significativa entre si quando comparados os resultados obtidos nas 5 bases diferentes. Esse teste consiste em verificar o quão bem os dados obtidos se encaixam em uma distribuição desejada [20], ou seja, o objetivo desse teste é verificar se os resultados obtidos das 5 diferentes bases de dados são consistentes entre si.

Z-test: Foi utilizado um teste de hipótese para verificar o efeito dos pré-processamentos comparado com o baseline (base sem tratamento). O objetivo desse teste é minimizar o efeito aleatório dos dados e determinar se um pré-processamento é estatisticamente impactante no resultado obtido [20]. Cada uma das métricas de avaliação obtidas foi submetida a esse teste a fim de verificar as variações encontradas. Foi utilizado o *Z-test* com um nível de significância de 0,05.

III. Resultados e Discussão

Para o treinamento dos modelos, foram consideradas as combinações de base, tipo de pré-processamento, modelo de vetorização, número de vetores e modelo de classificação, totalizando 560 resultados. Os modelos do NILC foram treinados apenas com o pré-processamento *base* (dados sem pré-processamento) e *min*, pois os outros formatos não fariam sentido para esse formato. Foram calculados para os dados de treino a acurácia e para a base de teste a acurácia, recall e precisão.

Para verificar a validade estatística dos dados, foi aplicado o teste qui quadrado nas cinco bases distintas de dados, verificando se as métricas calculadas estavam dentro de um intervalo estatístico confiável quando compartilhavam as outras informações iguais, de modo que todos os agrupamentos mostraram-se consistentes. Dessa forma, foi calculado a média aritmética das métricas para todas as bases, possibilitando consolidar os resultados dentro das combinações entre pré-processamento, modelo de vetorização, número de vetores e classificador.

O impacto do pré-processamento foi dimensionado calculando a variação da métrica analisada em relação ao *baseline*. A fim de estabelecer uma significância estatística nos resultados, um z-test foi aplicado para determinar se o tratamento gerou um impacto positivo, negativo ou nulo frente ao baseline. Além dos pré-processamentos, foram feitos os testes de hipóteses para validar estatisticamente o impacto do aumento do número de vetores, modelo de classificação e formato de vetorização (modelos W2V).

A. Impacto da configuração do processo nos resultados

Os modelos que utilizaram a vetorização como W2V tiveram comportamento semelhante. Para os dados do NILC, o formato da vetorização e o números de vetores não apresentaram uma variação estatística relevante. Quanto aos modelos, o classificador GNB obteve uma acurácia de teste próxima de 57% no baseline e o tratamento *min* aumentou apenas 0,6p.p. desta métrica. Os resultados obtidos pela LR foram superiores no baseline, próximos a 76% de acurácia de teste, mas o pré-processamento reduziu a métrica em 2,5 p.p.

Os modelos W2V treinados tiveram um comportamento similar, o número de vetores e o formato da vetorização não tiveram impacto nas métricas, mas sim o modelo de

classificação utilizado. Para os modelos submetidos ao GNB, todos os pré-processamentos tiveram pouco impacto nas métricas avaliadas. Por outro lado, os pré-processamentos dos modelos LR tiveram um efeito nas métricas, as bases submetidos a *ste* e *min* obtiveram, no geral, resultados positivos, assim como sua combinação, enquanto *stop* reduziu a acurácia de teste, de modo que as intersecções com este tratamento gerou um impacto negativo nos resultados. Um ponto a se destacar dessa vetorização é que todos os modelos tiveram um recall bem alto, sendo um indício que para este problema específico essas configurações identificam melhor comentários positivos do que negativos.

Nos modelos W2V, os dados treinados com GNB tiveram um resultado favorável para as vetorizações treinadas com os dados, enquanto modelos com LR obtiveram melhores resultados com as informações do NILC.

A vetorização TF-IDF obteve resultados diferentes: o número de vetores utilizados afetou o resultado — o maior número de vetores gerou um resultado melhor. Além disso, os dois classificadores tiveram acurácia de teste superior a 80%, sendo o segundo superior ao primeiro. O efeito dos pré-processamentos seguem a mesma lógica do W2V treinado com LR, enquanto *min* e *ste* afetam positivamente as métricas, *stop* reduz a acurácia dos modelos. Esses modelos obtiveram índices de recall aproximados, indicando que o modelo classifica os comentários de maneira mais homogênea.

B. Impacto dos pré-processamentos nos modelos

Para verificar o efeito dos pré-processamentos, foram excluídos os resultados obtidos pelo modelo do NILC, pois o único formato possível para testar seria o *base* e o *min*. Dessa forma, dos formatos possíveis, foram obtidos os resultados das métricas avaliadas conforme tabela 4.

Tabela 4. Efeito dos pré-processamentos em relação ao baseline.

Tratamento	*Var*	*min*	*stop*	*ste*	*min+ ste*	*min + stop*	*stop + ste*	*min + stop + ste*
Acurácia treino	Pos	5	0	6	10	1	0	4
	Nula	7	6	6	2	7	5	4
	Neg	0	6	0	0	4	7	4
Acurácia teste	Pos	5	1	6	10	1	0	5
	Nula	7	6	6	2	7	3	3
	Neg	0	5	0	0	4	9	4
Recall teste	Pos	4	4	3	4	4	4	5
	Nula	2	3	3	3	3	2	3
	Neg	6	5	6	5	5	6	4
Precisão teste	Pos	4	0	6	7	1	0	3
	Nula	8	6	5	5	5	7	4
	Neg	0	6	1	0	6	5	5
Total	Pos	18	5	21	31	7	4	17
	Nula	24	21	20	12	22	17	14
	Neg	6	22	7	5	19	27	17

Isoladamente a stemização foi o pré-processamento que gerou maior impacto positivo nas métricas, enquanto a remoção de *stop words* ocasionou maior volume de reduções nas métricas. Os resultados relacionados ao pré-processamento *stop* podem acontecer devido às palavras retiradas da análise serem importantes para a classificação dos comentários. Quando verificada a sinergia dos pré-processamentos, a combinação *min* + *ste* obteve o melhor resultado, aumento a acurácia de treino em 83% dos casos. Por outro lado, a combinação *stop* + *ste* obteve o pior desempenho, impactando negativamente a mesma métrica em 58% dos casos e sem ter efeito positivo em nenhum dos testes. A figura 2 mostra a proporção dos impactos na acurácia de treino.

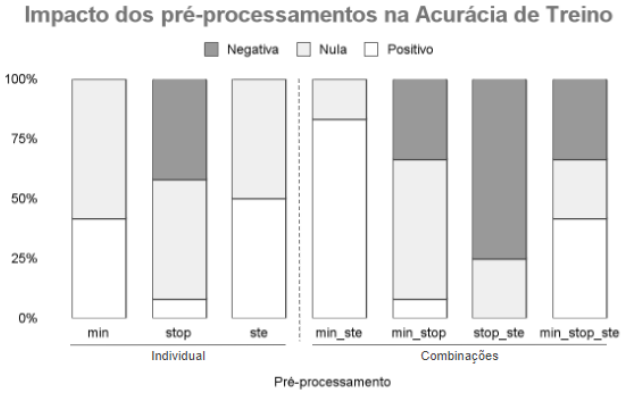


Fig 2. Impacto dos pré-processamentos na acurácia de treino.Fonte: Autor

C. Fatores de maior influência dos processos

Os incrementos dos pré-processamentos nos modelos W2V são baixos e pouco impactantes no resultado final, assim como o número de vetores utilizados e o formato da vetorização (CBOW ou Skip-gram). Neste tipo de vetorização, o impacto do classificador utilizado se mostrou mais significativo em relação às outras configurações.

Nos modelos utilizando TF-IDF como vetorização, embora exista um impacto do classificador e do número de vetores, o aumento é gradual seguindo o aumento de complexidade da configuração, obtendo um incremento próximo do gerado pelos pré-processamentos. Assim, nesse método de vetorização, os tratamentos utilizados na base apresentam uma maior relevância, pois geram incrementos da mesma magnitude da alteração das configurações, visto que os resultados obtidos com a base sem tratamento já possuem uma acurácia acima de 80%.

A figura 3 mostra um comparativo das combinações ilustrando a acurácia de teste e seu valor incremental em pontos percentuais quando aplicado o pré-processamento que obteve o melhor resultado. Cada modelo está representado por uma círculo no gráfico, onde a área menor representa os modelos com 100 vetores e a maior os de 300. Além disso, dentro da área pontilhada está a combinação de tipo de vetorização e modelo de classificação. As cores representam o pré-processamento que obteve o melhor resultado. Essa figura ilustra bem como o impacto do modelo de classificação é mais importante para os modelos baseados no W2V, além de mostrar o quão perto as acurácias de teste ficaram nesses modelos.

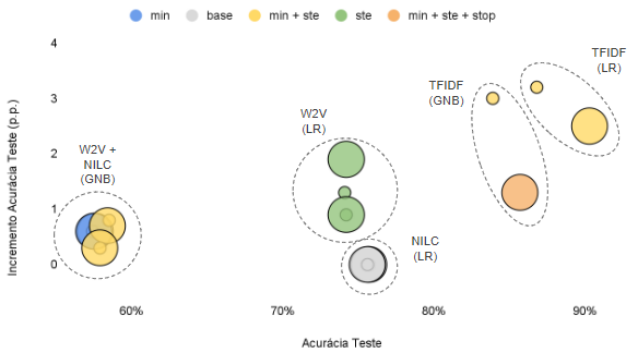


Fig 3. Distribuição dos melhores resultados em cada um dos modelos comparando a acurácia de teste e o incremental em ponto percentual da métrica.Fonte: Autor

A figura 4 e a tabela 5 mostram os melhores resultados considerando o melhor tipo de pré-processamento e o incremento do tratamento em relação ao *baseline*.

TABELA 5. MELHORES RESULTADOS POR PRÉ-PROCESSAMENTO

Vetorização	*Vetores*	*Modelo*	*Pré-pr.*	*Acur. Treino*	*Inc. Treino*	*Acur. Teste*	*Inc. Teste*
NILC CBOW	100	GNB	min	57,8%	0,4 p.p.	57,6%	0,6 p.p.
NILC CBOW	300	GNB	min	57,8%	0,4 p.p.	57,6%	0,6 p.p.
NILC CBOW	100	LR	base	76,1%	-	75,7%	-
NILC CBOW	300	LR	base	76,1%	-	75,7%	-
NILC SKIP	100	GNB	min	57,7%	0,5 p.p.	57,4%	0,6 p.p.
NILC SKIP	300	GNB	min	57,7%	0,5 p.p.	57,5%	0,6 p.p.
NILC SKIP	100	LR	base	76,0%	-	75,6%	-
NILC SKIP	300	LR	base	76,0%	-	75,6%	-
CBOW	100	GNB	min + ste	58,7%	0,7 p.p.	58,5%	0,8 p.p.
CBOW	300	GNB	min + ste	58,7%	0,7 p.p.	58,4%	0,7 p.p.
CBOW	100	LR	ste	74,0%	1,5 p.p.	74,1%	1,3 p.p.
CBOW	300	LR	ste	74,0%	1,7 p.p.	74,2%	1,9 p.p.
SKIP	100	GNB	min + ste	58,2%	0,3 p.p.	57,9%	0,3 p.p.
SKIP	300	GNB	min + ste	58,2%	0,4 p.p.	57,9%	0,3 p.p.
SKIP	100	LR	ste	74,1%	1,0 p.p.	74,2%	0,9 p.p.
SKIP	300	LR	ste	74,1%	1,0 p.p.	74,2%	0,9 p.p.
TF-IDF	100	GNB	min + ste	83,8%	2,8 p.p.	83,9%	3,0 p.p.
TF-IDF	300	GNB	min + ste + stop	85,7%	1,4 p.p.	85,7%	1,3 p.p.
TF-IDF	100	LR	min + ste	87,3%	3,4 p.p.	86,8%	3,2 p.p.
TF-IDF	300	LR	min + ste	90,1%	2,4 p.p.	90,3%	2,5 p.p.

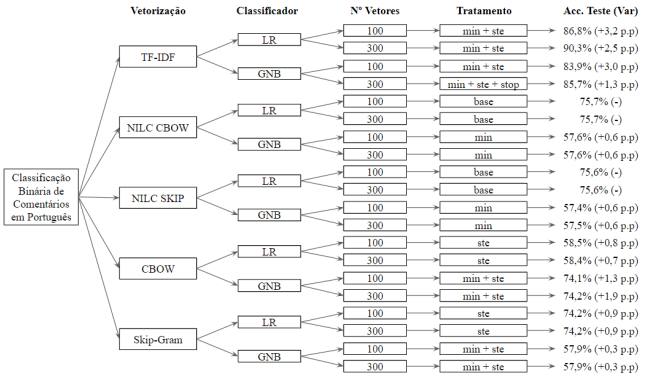


Fig 4. Melhor acurácia de teste considerando o pré-processamento com maior impacto positivo nas combinações de vetorização, classificador e número de vetores. Fonte: Autor

IV. CONCLUSÃO

Considerando um problema de classificação binária de comentários de marketplace brasileiros, os resultados obtidos mostram que a configuração do tipo de vetorização e o tipo de classificador possuem um impacto geral superior ao modo de pré-processamento dos dados. Embora possa existir um impacto nos resultados, o *pipeline* para resolução do problema tem maior relevância do que o pré-processamento isolado. Dentre os tratamentos individuais utilizados, o *ste* obteve o maior número de impactos positivos nas métricas e o *stop* teve influência negativa.

Quando comparado às demais combinações de pré-processamentos, *min* + *ste* apresentou melhores resultados, enquanto *stop* + *ste* gerou impactos negativos. Os modelos baseados em W2V são mais sensíveis ao tipo de classificador utilizado, porém a variação de vetores utilizados não teve um impacto significativo nos resultados.

O TF-IDF se mostrou mais robusto, de modo que o aumento de complexidade nos formatos foi gerando um incremento gradual nas métricas. A acurácia de teste mais alta foi obtida no TF-IDF com 300 vetores utilizando como classificador o LR e com *min* + *ste* de pré-processamento e o pior resultado foi o Skip-gram com 100 vetores utilizando o GNB e com *stop* + *ste* de tratamento.

REFERENCES

[1] S. Vajjala, B. Majumber, A. Gupta, H. Surana. "Practical Natural Language Processing: A comprehensive guide to building real-world NLP Systems"; Sebastopol, CA, USA: O'Reilly Media. 2020.

[2] H. Lane, H. Hapke, C. Howard."Natural Language Processing in Action: Understanding, analyzing, and generating text with Python." Shelter Island, NY, USA: Manning Publications Company, 2019.

[3] N. Mehta, D. Steinman, L. Murphy. "Customer Success: how innovative companies are reducing churn and growing recurring revenue." [S.l.]: John Wiley & Sons, 2016.

[4] P. Santana. "Consumer Insight: construindo experiências verdadeiramente centradas no cliente." São Paulo: Évora. 2018.

[5] N. Indurkhya, F. Damarau. "Handbook of natural language processing." 2ed. Florida: CRC Press, 2010.

[6] J. Brownlee. "Deep Learning for Natural Language Processing." Machine Learning Mystery, Vermont, Australia, 2017.

[7] A. Géron. "Hands-on machine learning with Scikit-Learn, Keras and TensorFlow." O'Reilly Media, Inc. 2022.

[8] Kaggle. "Brazilian Portuguese Sentiment Analysis Datasets". Created by Fred Dias. Disponível em: https://www.kaggle.com/datasets/fredericods/ptbr-sentiment-analysis-datasets. Acesso em: 12 nov. 2022. 2021a.

[9] Kaggle. "Brazilian E-Commerce Public Dataset by Olist. Version 2". Created by Francisco Magioli. Data Update: 2021/10/01. Disponível em: https://www.kaggle.com/datasets/olistbr/brazilian-ecommerce. Acesso em: 04 jun. 2022. 2021b.

[10] L. Real, M. Oshiro, A. Mafra. "B2w-reviews01 - an open product reviews corpus." STIL - Symposium in Information and Human Language Technology. Disponível em: https: //github.com/b2wdigital/b2w-reviews01. 2019.

[11] N. Hartmann, L. Avanço, P. Balage, M. Duran, M. Nunes, T. Pardo, S. Aluísio. “A large corpus of product reviews in portuguese: Tackling out-of-vocabulary words.” Em: Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14). 2014.

[12] F. Souza, J. Filho. “Sentiment Analysis on Brazilian Portuguese User Reviews.” Em: 2021 IEEE Latin American Conference on Computational Intelligence (LA-CCI). IEEE, 2021.

[13] J. Prusa, T. Khoshgoftaar, N. Seliya. “The effect of dataset size on training tweet sentiment classifiers.” Em: 2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA). IEEE, 2015.

[14] N. Hartmann, E. Fonseca, C. Shulby, M. Treviso J. Rodrigues, S. Aluísio. “Portuguese word embeddings: Evaluating on word analogies and natural language tasks.” arXiv preprint arXiv:1708.06025. 2017.

[15] A. Aizawa. “An information-theoretic perspective of tf-idf measures.” Information Processing & Management, V. 39, n1 p. 45-65, 2003.

[16] T. Mikolov, K. Chen, G. Corrado, J. Dean. “Efficient estimation of word representations in vector space.” arXiv preprint arXiv:1301.3781, 2013.

[17] M. T. Khan, M. Durrani, A. Armughan, I. Inayat, S. Khalid, K. H. Khan. “Sentiment analysis and the complex natural language.” Complex Adaptive Systems Modeling, 4(1), 1-19. 2016.

[18] F. Provost, T. Fawcett. “Data science para negócios: o que você precisa saber sobre mineração de dados e pensamento analítico de dados.” 1ª ed. Rio de Janeiro: Alta Books. 2016.

[19] G. H. S. Andreata. “O uso de processamento de linguagem natural para a análise de sentimentos na rede social Reddit.” Disponível em: https://repositorio.ucs.br/xmlui/bitstream/handle/11338/3804/TCC%20Guilherme%20Henrique%20Santos%20Andreata.pdf?sequence=1&isAllowed=y. Acesso em: 26 mai. 2022. 2018.

[20] A. Bruce, P. Bruce. “Estatística Prática para Cientistas de Dados." Alta Books, 2019.

Desenvolvimento de modelo para predição de cotações de ação baseada em análise de sentimentos de *tweets*

Mario Akita
Instituto Federal de São Paulo
Campinas, Brasil
mario.akita@aluno.ifsp.edu.br

Prof. Me. Everton Josue da Silva
Instituto Federal de São Paulo
Campinas, Brasil
everton.silva@ifsp.edu.br

***Abstract*—O treinamento de modelos de aprendizado de máquina para predição de cotações de ações tem sido um assunto cada vez mais abordado à medida que o avanço tecnológico possibilitou o envio automatizado e instantâneo de ordens de compra e venda desses ativos. Enquanto a grande maioria das abordagens nesta disciplina consiste em treinar modelos de Redes Neurais com base somente na cotação histórica dos ativos, neste trabalho utilizamos a plataforma *iFeel* 2.0 para extrair 19 indicadores de sentimentos de postagens da plataforma de *microblogs Tweeter* relacionadas à empresa Petrobras e treinamos modelos XGBoost para prever a cotação das ações desta empresa. Posteriormente, simulamos o desempenho deste modelo e comparamos à média de outros 100 aleatórios para determinar que houve ganho médio de R$88,82 (brutos) no período ao utilizar o modelo treinado, quando comparado ao rendimento médio dos outros cem modelos aleatórios.**

Keywords—análise de sentimentos, tweets, cotações, ações, Petrobrás, ifeel

I. INTRODUÇÃO

Desde a digitalização das operações de compra e venda de ativos financeiros intensificada nos anos 1990s, as operações de negociação de ações vêm sendo alvo de intenso estudo com o objetivo de gerar algoritmos que sejam capazes de trazer retorno financeiro aos investidores de maneira automatizada. Recentemente, dada a evolução do poder computacional que viabilizou modelos cada vez mais complexos de negociação, a negociação automática de ações através de algoritmos já representava um montante de pelo menos 50% de todas as negociações de ações nas bolsas de valores dos Estados Unidos no ano de 2012 [1].

Tradicionalmente, os modelos para predição de preços de ação são construídos com base em estatísticas sobre preços, volumes de negociação, médias móveis, dentre outras informações estatísticas e contábeis do ativo financeiro em questão, sendo considerados, portanto, uma evolução da escola de análise técnica de ações [2]. A recente disponibilização de grandes volumes de dados e incrementos no poder de processamento criou um ambiente propício para o desenvolvimento de novos algoritmos mais complexos[3] como diferentes redes neurais ou combinação de vários algoritmos clássicos que possibilitaram a utilização de outros aspectos não estatísticos como indicadores de sentimento em notícias [4], ou tweets [5] e [6].

Neste trabalho, utilizaremos as mais recentes técnicas de Processamento de Linguagem Natural (NLP) para extrair 19 indicadores de sentimentos através de modelos já existentes na literatura que são, posteriormente, utilizados como *features* juntamente com estatísticas de preços e volumes de negociação para treinamento de modelo computacional XGBoost com o objetivo de prever cotações futuras das ações preferenciais da empresa Petróleo Brasileiro S.A. – Petrobras (PETR4).

II. OBJETIVOS

Neste projeto, desenvolvemos modelos computacionais para prever variações nos preços da ação preferencial da Petrobras (PETR4). O objetivo do projeto é desenvolver modelos que apresentem melhor desempenho quando comparados a modelos aleatórios e que consigam melhores ganhos financeiros do que a realização de operações ao acaso dentro do intervalo de tempo reservado para testes.

III. BASES TEÓRICAS

A. iFeel 2.0

Aplicação Web implementada por Araujo *et al*. [7] para simplificar a implementação e o uso de múltiplos métodos de análise de sentimentos no nível de sentenças. São suportados 19 modelos de análise de sentimentos no total brevemente explicados a seguir:

1) Emoticons: Proposto por Gonçalves *et al*. [8] atribui uma pontuação de sentimentos baseado nos *emoticons* utilizados dentro da frase.

2) Happiness Index: Proposto por Dodds *et al*. [9], consiste em uma escala de 1 a 9 em que frases são classificadas de acordo com um uso de 1034 palavras e suas escalas na Affective Norms for English Words (ANEW) [10].

3) SentiWordNet: É uma ferramenta proposta por Esuli *et al.* [11] comumente utilizada na classificação de opiniões baseada em um dicionário léxico chamado WordNet que considera palavras em língua inglesa. O modelo agrupa palavras em conjuntos chamados de *synsets*. Em seguida, de acordo com as palavras e intensificadores deste conjunto, calcula uma pontuação para considerar o sentimento positivo ou negativo de cada *synset*. Ao final, todos os *synsets* são ponderados para calcular a polaridade global da sentença.

4) Senticnet: Proposto por Camnria *et al.* [12], é um modelo que utiliza técnicas de NLP baseado em inteligência artificial. Contém 14 mil conceitos que são utilizados para calcular a polaridade da sentença e foi inicialmente utilizado para avaliar os comentários de pacientes do National Heath System (NHS) na Inglaterra.

5) PANAS-t: É uma escala psicométrica proposta por Gonçalves *et al.* [13] para detectar humor baseado no

método PANAS (*Positive Affect Negative Affect Scale*) que analisa o texto diante de nove categorias de humor. Na implementação do *iFeel*, para gerar a escala de polaridade global a nível de sentença, foram considerados 4 humores como polaridade positiva, quatro como polaridade negativa e um neutro.

6) Sentistrength: Este modelo [14] propõe uma mistura de uma série de métodos de classificação supervisionados ou não (como regressão logística, *Support Vector Machine* (*SVM)*, árvores, dentre outros) para avaliar a polaridade do texto utilizando uma extensão do dicionário LIWC.

7) SASA: O *SailAil Sentiment Analyzer*[15] é baseado em técnicas de aprendizado de máquinas similares ao *SentiStrength* e foi desenvolvido para analisar postagens do *Twitter*. Na implementação do *iFeel*, foi utilizado um classificador *Naive Bayes* treinado pelos autores do método.

8) Opinion Lexicon: Focada originalmente em avaliações de produtos, o modelo *Opinion Lexicon* proposto por Hu *et al.* [16] construiu um dicionário léxico para capturar se as frases utilizadas em avaliações de produtos na internet eram positivas ou negativas

9) Opinion Finder (MPQA): Focado em identificar aspectos subjetivos das frases utilizando análise léxica e modelos de aprendizado de máquina. Foi proposto originalmente por Wilson *et al.* [17] e [18].

10) AFINN: Construído a partir de postagens da rede do *Twitter* para melhor capturar o linguajar utilizado na plataforma, o dicionário léxico proposto por Nielsen [19] é considerado uma expansão do dicionário ANEW[20] que propunha atribuir pontuações a palavras em inglês de acordo com a emoção a que elas eram associadas.

11) SO-CAL: Proposto por Taboada *et al.* [21], cria um dicionário léxico que contém n-gramas associados a uma escala de emoções que varia de -5 a +5 de modo a capturar palavras que intensificam o sentido de outras ou expressões ao invés de classificar palavras isoladamente.

12) Emoticons Distant Supervision: Criado a partir de uma ampla base de tweets, o modelo proposto por Hannak *et al.* [22] calcula a polaridade através do cálculo da frequência em que cada componente léxico aparece na frase analisada.

13) NRC Hashtag: O modelo proposto por Mohammad [23] pontua frases de acordo com a frequência em que determinadas *hashtags* são utilizadas na frase.

14) Emolex: Constrói um dicionário léxico associado a 8 emoções básicas. Utiliza unigramas e bigramas para associar as pontuações e foi proposto e construído colaborativamente por Mohammad *et al.* [24].

15) SANN: Criado para recomendação de conteúdos multimídia no *TED Talks*, o modelo [25] utiliza ações do usuário como indicadores para inferir sentimentos dos usuários o os combina com comentários não classificados para gerar um modelo *Sentiment-aware nearest neighbour model* (SANN).

16) Sentiment140 Lexicon: Utiliza um dicionário léxico calculado com base no *dataset* utilizado para treinar o método *Sentiment140*. Proposto por Mohammad *et al.* [26], a pontuação de cada expressão contida no dicionário foi calculada levando em conta a utilização de *emoticons* em *tweets* e a frequência de utilização de uma expressão nos *tweets* de determinada classe.

17) Stanford Recursive Deep Model: Introduz um modelo *Recursive Neural Tensor Network* (RNTN) que computa cada frase de acordo com a maneira em que seus componentes interagem entre si dentro da frase. Proposta por Socher *et al.* [27], utiliza recursividade para levar em consideração, por exemplo, a posição em que cada palavra ou expressão parece na frase na hora de calcular a polaridade de cada frase.

18) Umigon: O modelo proposto por Levsallois [28] utiliza heurísticas para detectar negações, palavras alongadas e avalia *hashtags* com o objetivo de desambiguar frases.

19) Vader: Validado por humanos, o método proposto em 2014 [29] foi criado a partir de dados do *Twitter* e outras redes sociais. Utiliza uma abordagem que valoriza a opinião humana uma vez que se utilizou de avaliadores humanos e coleta de opinião de massas em seu desenvolvimento.

B. XGBoost

É um método para treinamento de árvores de decisão (*decision trees*) baseado em aumento do gradiente (*gradient boosting*) [30]. Muito utilizado em competições de aprendizado de máquina, possui uma performance comparável ao estado da arte estando presente, por exemplo, em 17 das 29 soluções vencedoras de desafios da plataforma *Kaggle* em 2015 [31].

Foi escolhido para o treinamento dos modelos deste trabalho por apresentar grande versatilidade e bom desempenho na solução de diversos problemas de aprendizado de máquina e pela experiência anterior em utilizá-lo para predição de preços de ações.

IV. METODOLOGIA

Todos os scripts foram desenvolvidos em *python* 3 utilizando notebooks da plataforma *jupyter* nos ambientes *anaconda* 3 instalados em computador pessoal e através da plataforma *Google Colab PRO* (exceto a tradução dos *tweets*, que foi realizada no *Google Spreadsheets*).

Um desenho esquemático do *pipeline* implementado pode ser observado na Fig 1.

A. Aquisição de cotações históricas de ações

As cotações para as ações preferenciais da companhia Petróleo Brasileiro AS – Petrobras (PETR4) foram obtidas através da plataforma *Meta Trader 5*, disponibilizada gratuitamente para clientes da *Clear* Corretora (XP Investimentos CCTVM S.A.). Por questões operacionais existe um limite de dados disponíveis na plataforma e, para este estudo, foram obtidos dados referente ao período de 3/12/2018 a 28/11/2022. Os dados obtidos têm granularidade de 5 minutos.

O dataset obtido é composto pelos seguintes campos:

- **Date**: Data do período considerado.
- **Time**: Hora de início do período considerado.
- **Open**: Preço da primeira negociação do período.
- **High**: Maior preço negociado no período.
- **Low**: Menor preço negociado no período.

- **Close**: Preço da última negociação do período.
- **Tickvol**: Quantidade de negócios realizados no período considerado.
- **Vol**: Quantidade de ações negociadas no período.
- **Spread**: Indica se houve ou não *spread* (diferença no preço de compra e venda) nos negócios do período.

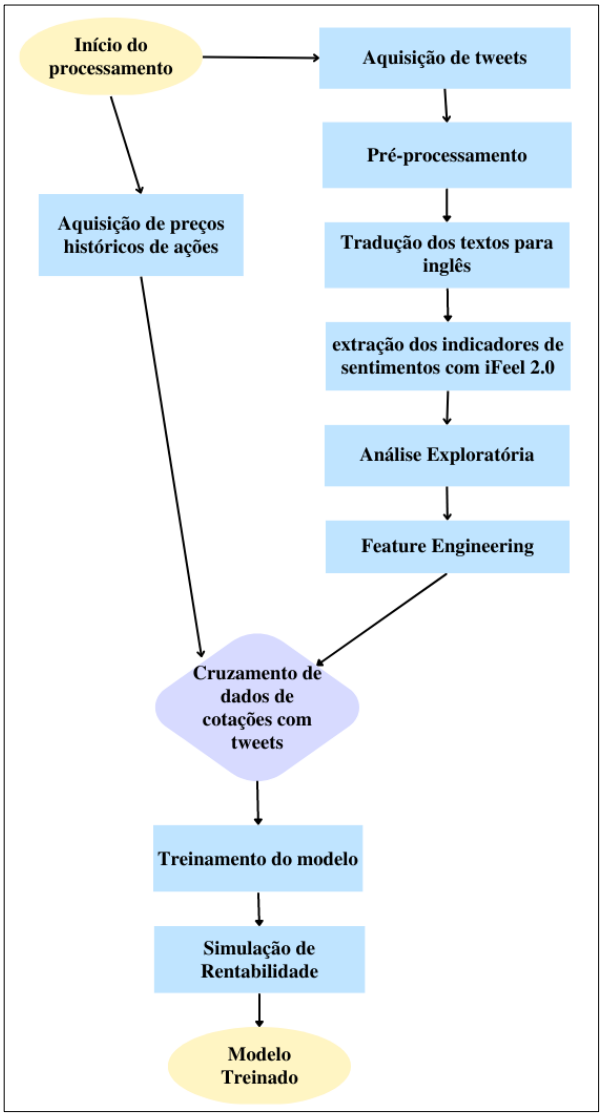


Fig. 1. Representação esquemática do pipeline implementado neste trabalho.

B. *Aquisição de tweets*

A aquisição de *tweets* foi realizada através de script em *Python* desenvolvido pelo autor. Os dados foram obtidos através da API v2 oficial do *Twitter* [33] utilizando a biblioteca *Tweepy*. Foi concedido o status de "pesquisador" para o uso acadêmico da API que possibilitou o incremento da quantidade de *tweets* consultados.

Para construção da base, utilizamos, inicialmente, todos os *tweets* postados entre 13:30 e 19:50 GMT – o que corresponde ao horário habitual de negociação das ações na Bolsa de Valores brasileira – do período de 23/8/2021 a 30/6/2022. Posteriormente, a base foi aumentada para conter os *tweets* criados entre 1/1/2021 e 30/11/2022.

A consulta filtrou apenas tweets que continham as *hashtags* #PETR3 ou #PETR4, além daqueles que continham o nome da empresa "Petrobras". Foram excluídos os *retweets* e aqueles escritos em outra língua que não o português.

Os dados contidos no *dataset* de *tweets* são os seguintes:

- **Created_at**: *timestamp* correspondente à data e hora em que a postagem foi publicada.
- **Text**: o texto da postagem.
- **Like**: quantidade de "curtidas" da postagem.
- **Quote**: quantidade citações da postagem.
- **Reply**: quantidade de respostas da postagem.
- **Retweet**: quantidade de usuários que republicaram a postagem em suas contas.
- **User_followers**: quantidade de seguidores do usuário autor da postagem.
- **User_following**: quantidade de usuários que o autor da postagem segue.
- **User_tweets**: quantidade total de postagens do autor.
- **User_listed**: quantidade de listas criadas por outros usuários que contém o autor da postagem.

Devido ao limite imposto pela API do *Twitter* de retorno de, no máximo, 300 *tweets* por consulta, 180 consultas a cada 15 minutos e uma consulta por segundo, houve a necessidade de dividir as consultas por período de tempo e retardar o processamento das consultas utilizando o comando *sleep()* da linguagem *Python* e limitar manualmente as consultas para que apenas um mês fosse consultado a cada execução do *script*.

C. *Pré-processamento dos tweets*

Após a consulta dos *tweets* postados nos períodos pretendidos, foram efetuadas operações para pré-processar os dados obtidos e retirar alguns elementos que não seriam utilizados.

A primeira medida tomada foi a filtragem dos *links* presentes nas postagens. Por se tratarem apenas de endereços de *sites*, foram filtrados pois não influenciam no cálculo de polaridade dos *tweets*. Outro elemento eliminado das bases foi a menção a outros usuários. Menções são caracterizadas pelo caractere "@" seguido do nome de algum usuário da rede e servem como uma espécie de *link* para marcar outros usuários em postagens que podem interessar a eles. Foram filtrados pois não representam informação relevante para o cálculo da polaridade dos *tweets*.

Foram excluídos, ainda, símbolos não alfabéticos, *emojis* e pontuações. A ideia por trás dessa eliminação foi tentar facilitar obtenção das métricas de polaridade dos *tweets* em um contexto de desenvolvimento de modelos de NLP próprios específicos para este trabalho (como originalmente pensado). Entretanto, o simples desenvolvimento de um modelo de linguagem é uma tarefa complexa que exige quantidades massivas de dados para o treinamento consequentemente a ideia do desenvolvimento acabou substituída. Porém, uma vez que o dataset já havida sido processado quando da mudança de curso, os *emojis* e símbolos removidos acabaram comprometendo o cálculo da polaridade em alguns dos modelos utilizados.

Outro processamento efetuado foram a separação de data e hora e conversão para o fuso horário brasileiro (GMT-3). Necessária para a correta atribuição dos preços e estatísticas obtidos no item anterior aos tweets, os procedimentos visam a padronização de datas e horas para evitar erros e simplificar visualização do pipeline construído. Outra operação efetuada foi a eliminação de duplicatas, que nada mais é do que um procedimento de saneamento de *datasets* corriqueiro, mas importante para evitar contaminação dos dados quando divididos em treinamento e teste. Sua falta poderia prejudicar o cálculo das estatísticas de cada período e, consequentemente, o treinamento do modelo como um todo.

Finalmente, foram eliminados os *tweets* com pouca informação. Postagens com 2 palavras ou menos assim como aquelas com menos de 20 caracteres foram eliminadas pela alta chance de não conterem informações relevantes.

D. Tradução do texto dos tweets

Devido à carência de modelos treinados especificamente para o idioma português brasileiro, o sistema *iFeel* 2.0 conta com uma ferramenta de tradução embutida. Apesar disso, o módulo de tradução apresentava erro de conexão com a API e estava indisponível nos períodos de testes. Diante deste cenário, a tradução dos textos dos *tweets* que foram extraídos foi alcançada através da utilização das funções integradas ao *Google Spreadsheets*.

Esta função utiliza internamente o *Gooogle* Tradutor para prover as traduções e foi necessária uma vez que todos os modelos utilizados pelo iFeel 2.0 utilizam-se de dicionários léxicos em inglês ou foram treinados utilizando-se bases textuais naquele idioma.

E. Extração dos indicadores de sentimentos

A extração das polaridades de sentimentos foi executada utilizando o programa *iFeel* 2.0. Disponível para *download* em uma imagem *Docker*, o programa foi executado mês a mês em três ambientes: um computador portátil pessoal, um computador pessoal de mesa e um servidor em nuvem.

O processamento se mostrou desafiador, já que congelamentos e travamentos da máquina virtual Java utilizada no projeto ocorreram frequentemente e necessitaram de constante intervenção manual. Ao final deste processo, os *datasets* mensais foram combinados para formar um grande conjunto de dados de 323.460 amostras e 39 atributos que abrangem conteúdo publicado em um intervalo de tempo de 22 meses.

F. Análise exploratória

Após a extração de dados, foi efetuada uma breve análise exploratória de dados para melhor compreensão do conjunto de dados a ser trabalhado. A seguir, alguns dos principais achados são discutidos:

1) Balanceamento de classes: Há um leve desbalanceamento de amostras. Enquanto são 139.885 as amostras para períodos de alta nas cotações (~56,5% do total), as amostras referentes a períodos baixa somam 107.718 amostras (~43,5% do total).

2) Distribuição dos atributos de sentimentos: De maneira geral, a grande maioria dos tweets foram classificados como neutros com as demais postagens se distribuindo de maneira razoavelmente equilibrada dentre as duas polaridades. Há exceções como o método *PANAS-t* que classificou quase todas as amostras como neutras, *EMOLEX* e *OPINION LEXICON* que classificaram muito mais amostras como negativas. Outros modelos, como *EMOTICONS* e *EMOTICONS*-DS foram, obviamente, prejudicados pela limpeza de *emoticons* realizada no pré-processamento. A fig 2 traz a distribuição de algumas das *features* extraídas.

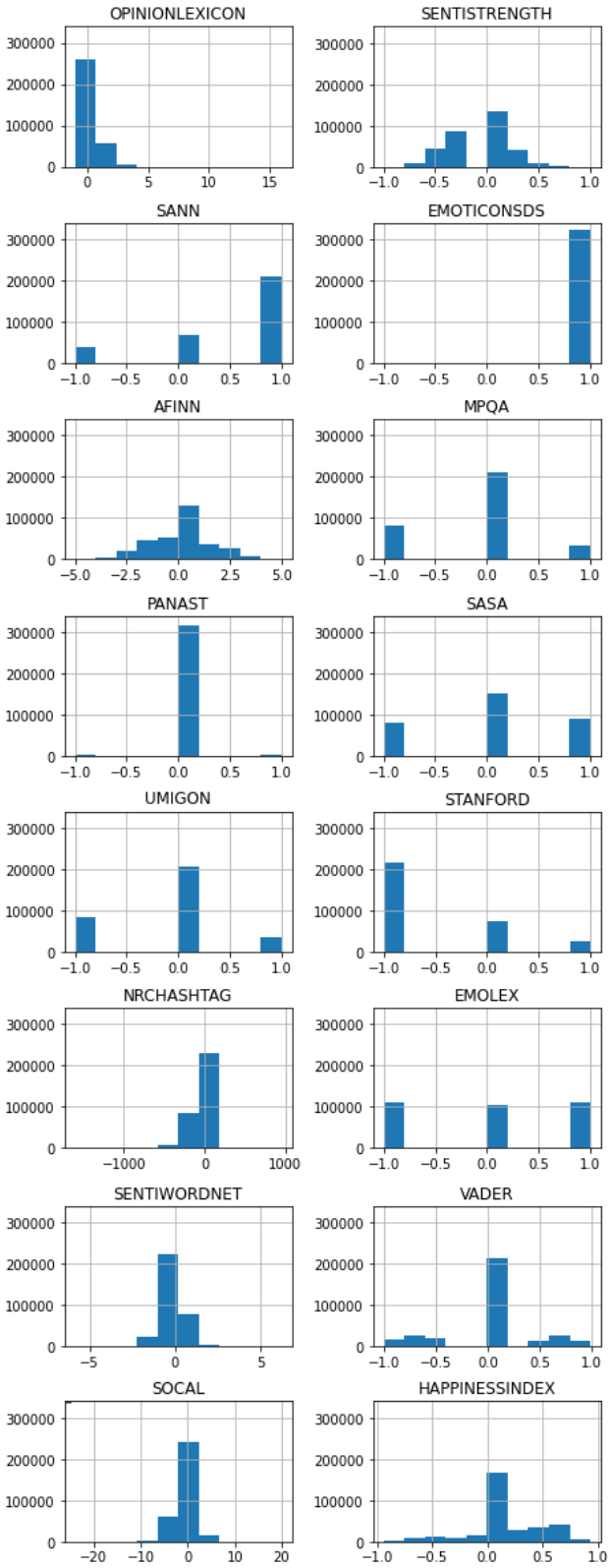


Fig. 2. Distribuição de pontuação das amostras de alguns modelos.

3) Estudo de correlações: A variável-alvo não apresentou correlação significativa com nenhum dos outros atributos sendo que as únicas correlações importantes foram registradas entre os valores de polaridade calculados – o que, de certa forma, é esperado uma vez que muitos modelos podem ser treinados a partir de bases textuais similares.

G. Feature engineering

Nesta etapa, com o intuito de aumentar a quantidade de *features* no dataset, foram criados atributos com o intuito de prover outras informações que talvez fossem relevantes durante o treinamento, como a hora de criação e tamanho dos *tweets*. A ideia é aumentar a variabilidade de *features* a serem apresentadas para o treinamento com outros dados que não somente aqueles referentes à análise de sentimentos.

1) Hora: número inteiro correspondente à hora em que a postagem foi publicada.

2) Word_count: contagem de palavras na publicação. Foi utilizada, também, para uma limpeza de dados que eliminou todas as postagens com menos de 3 palavras.

3) Text_length: tamanho total do *tweet*. Pensada para ser utilizada como um indicador de confiabilidade das *features* de polaridade durante o treinamento, parte do princípio de que quanto maior a publicação, maior a quantidade de palavras consideradas no cálculo da polaridade e, por consequência, mais "confiável" é a métrica.

Para formar o *dataset* final, foram utilizadas estatísticas de todos os *tweets* contidos em cada intervalo para que cada intervalo de tempo fosse representado por uma única amostra de dados, e, desta forma, criar uma base balanceada em número de amostras por período e inferir uma sequência entre as amostras de cada período.

Para cada um dos 27 atributos dos tweets e polaridades calculadas, foram utilizadas as seguintes estatísticas:

- Média
- Desvio-padrão
- Mínimo
- Máximo
- Soma
- Variância
- Contagem da quantidade de amostras do período

Além das estatísticas, foram adicionadas as *features* com defasagem destas estatísticas e dos atributos de preços e volumes de negociação. Desta forma, criou-se uma relação temporal entre os dados, o que é importante pois durante o treinamento o modelo será exposto a dados do tempo atual assim como dados dos tempos anteriores.

Foram utilizadas as seguintes janelas de tempo para adição dos atributos com atraso:

- Atraso de 5 minutos
- Atraso de 10 minutos
- Atraso de 15 minutos
- Atraso de 20 minutos

H. Cruzamento de dados de cotações com tweets

Devido às mudanças de estratégias ocorridas durante o desenvolvimento do projeto, este passo foi realizado em diferentes pontos do *pipeline* de dados executado. Apesar disto, por se tratar de um cruzamento de dois conjuntos de dados, ocorre apenas a adição de novas *features* para cada amostra, e sua presença ou ausência não interfere nos processamentos das etapas anteriores.

Este passo busca atribuir os preços e estatísticas de negociações da ação com os *tweets* que foram publicados naquele momento, portanto, o horário de postagem de cada *tweet* é arredondado para o múltiplo de 5 imediatamente inferior e as cotações desse tempo são atribuídas à postagem.

Adicionalmente, o conjunto de dados também recebe seu atributo-alvo. Como o objetivo do trabalho é prever a cotação da ação no próximo período de tempo (5 minutos), atribuímos o preço de fechamento em t + 5min a cada amostra para utilizarmos como alvo durante o treinamento.

I. Treinamento dos modelos

Foi treinado um modelo XGBoost para realizar todas as previsões do dia utilizando os seguintes parâmetros de treinamento:

- ETA: 0.01
- N_ESTIMATORS: 300
- RANDOM_STATE: 4321
- SCALE_POS_WEIGHT: 0,6
- MAX_DEPTH: 5
- OBJECTIVE: "binary_logistic"

1) Separação entre conjunto de teste e validação

Antes de definir quais os dados serão considerados na hora da divisão, é necessário definir a quantidade de dias a serem incluídos em cada grupo. Por padrão, foram utilizados os seguintes valores:

- Dias de treinamento: 200
- Dias de validação: 1
- Dias de teste: 1

2) Treinamento do modelo de treinamento

Desta forma, por exemplo, para treinar o primeiro modelo de validação, no *dataset* criado, utilizaríamos, como dados de treinamento, todas as amostras compreendidas entre 4/1/2021 e 26/10/2021. As amostras do dia seguinte (27/10/2021) são utilizadas para validação de forma a otimizar o modelo para este período. Note-se, entretanto, que os dados a serem de fato classificados são os do dia 28/10/2021 – que não são utilizados neste primeiro treinamento.

3) Treinamento do modelo de validação

Para gerar o modelo que fará a previsão de dados, retreina-se o melhor modelo obtido no passo anterior com dados dos dias de treinamento adicionados dos dados do dia de validação. O desempenho somente é aferido utilizando os dados do dia reservado para testes.

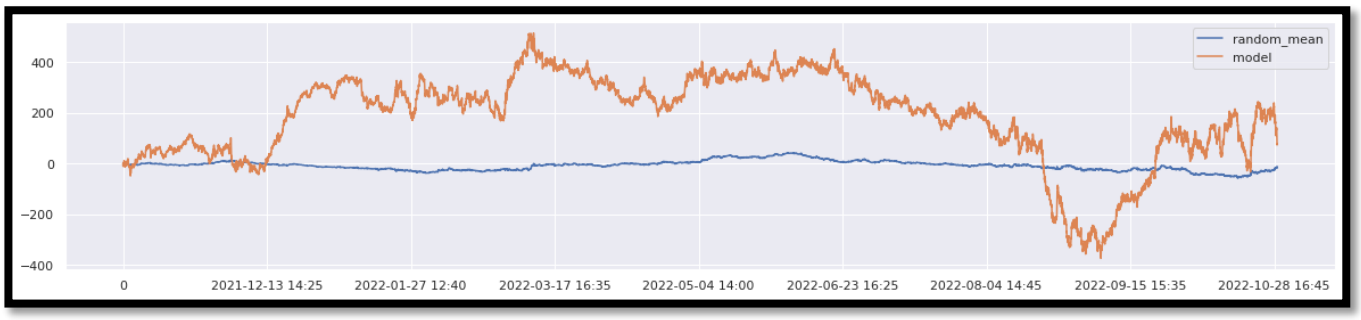


Fig. 3. Gráfico mostrando a rentabilidade no período de testes.

J. *Simulação de desempenho*

Foi implementado um calculador de desempenho para verificar qual seria o resultado financeiro ao se aplicar o algoritmo de predição treinado simulando operações de compra e venda da ação, de acordo com o resultado previsto pelo modelo. Foi utilizado um conjunto de 100 modelos aleatórios como *baseline* para entender se o modelo treinado realmente produz melhores resultados quando comparado à média dos modelos aleatórios.

Para tanto, foi utilizada a própria tabela de cotações e as seguintes regras:

- Caso o valor predito pelo modelo (ou o valor escolhido aleatoriamente, para os modelos aleatórios) seja 1, é simulada uma compra da ação no preço de abertura seguida de uma venda ao final do período.
- Caso contrário faz-se uma venda descoberta do papel, que consiste na venda do papel pelo preço de abertura seguida de uma compra ao final do período.
- Calcula-se a rentabilidade diária de cada modelo de acordo com a tabela a seguir:

TABLE I. CÁLCULO DA RENTABILIDADE DIÁRIA

Previsão	Fórmula
0	$Rentabilidade = QTDE * (P_{CLOSE} - P_{OPEN})$
1	$Rentabilidade = QTDE * (P_{OPEN} - P_{CLOSE})$

[a] P refere-se ao preço do ativo na abertura (OPEN) e fechamento (CLOSE)

Para todos os efeitos de cálculo de rentabilidade, foram desconsideradas eventuais taxas de corretagem, emolumentos, impostos, juros e quaisquer outras cobranças. A rentabilidade final foi calculada sempre considerando a negociação de 1 lote de ações (100 ações).

K. *Apresentação e Análise de resultados*

O modelo treinado apresentou excesso de rentabilidade bruta de R$77,00 frente à média de -R$11,82 dos modelos aleatórios durante o período de testes entre 27/10/2021 e 28/10/2022. Ou seja, no período considerado houve um ganho médio de R$ 88,82 ao escolher o modelo treinado em detrimento de um modelo aleatório hipotético aqui representado pela média de 100 modelos aleatórios.

Um gráfico com a rentabilidade bruta obtida em cada período pode ser observado na figura 3. Nele, podemos observar que na maioria do período estudado, o modelo treinado esteve acima do modelo aleatório médio. Isto evidencia uma relativa consistência do modelo treinado em entregar os resultados melhores que o aleatório.

A seguir apresentamos as métricas *Macro* do modelo treinado durante o período de validação:

- Precisão: 0,51
- *Recall*: 0,52
- F-1: 0,40
- AUC: 0,5153
- *Logloss*: 0,6923

L. *Breve discussão de resultados*

Apesar das métricas *macro* não serem consideradas boas, o modelo proposto conseguiu obter algum resultado positivo frente a um conjunto de modelos operando aleatoriamente, o que demonstra uma certa consistência do modelo em entregar resultados diante do intervalo de tempo relativamente grande (cerca de 1 ano).

Quando consideramos os períodos em que o modelo ganhou ou perdeu da média dos modelos aleatórios, podemos observar que o modelo treinado ganhou em 215 períodos e perdeu em 37 deles.

Houve um lucro bruto calculado de R$ 77,00 durante o período reservado para testes, o que representa um ganho médio de R$0,31 por operação, considerando a compre e venda de um lote de ações (100 ações). A média dos cem modelos que operaram aleatoriamente, gerou um ganho total no período de R$ -11,82, o que equivale um ganho médio de R$ -0,05 por operação.

Algumas conjecturas de fatores que podem ter interferido com o desempenho do modelo são:

- Dados talvez não tenham correlação suficiente com a cotação de ações específicas: A plataforma *Tweeter* em geral, tende a ser cenário de disputas e reclamações, o que nos levaria a ter muito mais conteúdo de opiniões negativas do que positivas e isto nem sempre condiz com as cotações de ações. Este fenômeno pode ter acontecido já vez que apesar do *dataset* de preços ser levemente desbalanceado para o campo positivo, as polaridades são levemente mais negativas que positivas. Além disso, situações como o aumento do preço de combustíveis, por exemplo, podem aumentar a quantidade de críticas nas redes sociais enquanto as ações sobem, pelo possível aumento no faturamento e, posteriormente, nos lucros da empresa.
- Período eleitoral: Parte considerável dos dados de testes se deu em período eleitoral, o que pode ter

gerado um descolamento entre os dados coletados e as cotações já que a Petrobrás é uma empresa estatal. Este fato é melhor observado no período em que o desempenho do modelo ficou abaixo da média dos modelos aleatórios já na parte final do gráfico da fig. 3. O período (agosto/setembro de 2022) coincide exatamente com o início das campanhas eleitorais para as eleições gerais daquele ano.

- Tradução: A tradução automática dos tweets de português para a língua inglesa pode ter provocado a perda de conteúdo semântico e prejudicado o cômputo das polaridades, uma vez que expressões comumente utilizadas no Brasil podem ter carga sentimental maior ou menor do que suas equivalentes em inglês.

V. CONCLUSÃO E TRABALHOS FUTUROS

Considerando toda a extensão do trabalho desenvolvido até aqui, que contemplou desde a aquisição de dados passando pela extração de características, amplo estudo sobre os dados adquiridos até o treinamento e mensuração de resultados, podemos afirmar que foi um trabalho desenvolvido de ponta-a-ponta utilizando apenas recursos disponíveis na internet gratuitamente (exceto *Google Colab PRO*).

Neste sentido, o objetivo de desenvolver um modelo que pudesse superar o aleatório para negociação de ações baseada em indicadores de sentimentos de *tweets* foi considerado alcançado.

Há, entretanto, uma vasta quantidade de pontos em que futuros trabalhos e abordagens alternativas poderão corrigir ou melhorar as falhas cometidas na abordagem aqui descrita e, consequentemente, melhorar os resultados. A seguir, algumas sugestões para futuras continuações deste trabalho.

A. Expansão do intervalo de tempo analisado

Inicialmente pensado para cobrir uma granularidade maior de dados (intervalo de 1 minuto), encontrar dados disponíveis gratuitamente ou com preço muito baixo para este intervalo de tempo mostrou-se uma tarefa infrutífera. Entretanto, ao migrarmos para o intervalo de 5 minutos, pudemos achar com facilidade dados até o ano de 2018. Uma das ideias é utilizar essa base maior para adquirir mais dados de períodos não considerados neste trabalho e, eventualmente, gerar modelos melhores.

B. Expansão para ações de outras empresas

Neste trabalho focamos nas ações da empresa Petrobras por ser a ação mais líquida e, por ampla margem, a empresa negociada mais comentada da rede social – principalmente após os sucessivos reajustes de preços ocorridos em 2021 e 2022. Entretanto, a mesma estratégia poderia ser implementada para outras empresas (ou mesmo um conjunto de empresas) em novos trabalhos, aproximando-se da abordagem de publicações que utilizaram índices de ações[5].

C. Utilização de outras estratégias de modelagem

Neste trabalho focamos na utilização do *XGBoost* como algoritmo de preferência para treinarmos o nosso modelo, porém, há outros algoritmos que também poderiam ser implementados, como por exemplo:

1) LSTM: Muito utilizado para predição de dados que possuam um componente temporal entre as amostras[34], poderia ser utilizado para predição de cotações de ações;

2) SOFNN: Rede neural que utiliza lógica *Fuzzy* já foi utilizada para prever índices de ações através de dados obtidos com análise de sentimentos de postagens do *twitter* [5] e poderia ser aplicado na resolução deste problema.

D. Utilização de outras fontes de dados:

Outras redes sociais ou mesmo *feeds* de jornais e notícias poderiam ser utilizados para adicionar confiabilidade aos dados.

E. Utilização de outra função para medir o desempenho do algoritmo no treinamento

Como pudemos observar na apresentação dos resultados, nem sempre períodos de maior acerto ou mesmo maior AUC apresentam os melhores resultados. Por padrão, a biblioteca *XGBoot* vem com duas funções para auferir o desempenho durante o treinamento: *AUC* e *LogLoss*. Talvez fosse necessária a adição de algum componente de peso a estas funções ou desenvolvimento de função baseada na rentabilidade para melhorar as métricas de rentabilidade dos modelos gerados.

REFERÊNCIAS

[1] The New York Times, **Times Topics: High-Frequency Trading**, 2012.

[2] TRELEAVEN P., GALAS M, LALCHAND, Vidhi. **Algorithmic Trading review**. In: Communications of the ACM, Volume 56, issue 11. November 2013, pp 76-85.

[3] SHAH, Dev, HARUMA Isah, ZULKERNINE, Farhana. **Stock Market Analysis: A Review and Taxonomy of Prediction Techniques.** In: International Journal of Financial Studies. 2019. Disponível em: https://www.mdpi.com/2227-7072/7/2/26/pdf?version=1560755357. Acesso em 27de abril de 2022.

[4] DUARTE, João Victor G. **Análise e previsão de tendências do mercado financeiro brasileiro (B3) baseada em análise de sentimentos de notícias**. 2021. p. 39. Instituto Federal de Educação, Ciência e Tecnologia de São Paulo. Campinas, SP. 2021

[5] BOLLEN, Johan, Huina Mao; ZENG, Xiao-Jug. **Twitter mood predicts the stock market**. 2010. arXiv. https://arxiv.org/abs/1010.3003

[6] COLIANNI, Stuart; ROLASES, Stephanie; SIGNOROTTI Michael 2015. **Algorithmic Trading of Cryptocurrency Based on Twitter Sentiment Analysis**. Disponível em: http://cs229.stanford.edu/proj2015/029_report.pdf. Acesso em 20/04/2022.

[7] ARAUJO, Matheus et al. **iFeel 2.0: A multilingual Benchmarking System for Sentence-Level Sentiment Analysis.** In: Tenth International AAAI Conference on Web and Social Media. Brasil, 2016

[8] P. Gonçalves, M. Araújo, F. Benevenuto, and M. Cha**. Comparing and combining sentiment analysis methods**. In COSN, 2013.

[9] DODDS, P. S, DANFORTH, C. M. **Measuring the happiness of large-scale written expression: songs, blogs, and presidents.** Journal of Happiness Studies, 11(4):441–456, 2009.

[10] BRADLEY, M. M. LANG, P. J. **Affective norms for English words (ANEW): Stimuli, instruction manual, and affective ratings.** Technical report, University of Florida, 1999.

[11] ESULI, Andrea. SEBASTIANI, Fabrizio. **Sentwordnet: A publicly available lexicalresource for opinion mining**. In LREC, 2006.

[12] CAMBRIA, Erik, SPEER, Robert, HAVASI, Catherine, HUSSAIN, Amir. **Senticnet: A publicly available semantic resource for opinion mining**. In AAAI Fall Symposium Series, 2010.

[13] GONCALVES, Pollyana. BENEVENUTO, Fabrício, CHA, Meeyoung. **Panas-t: A pychometric scale for measuring sentiments on twitter.** CoRR, 2013.

[14] THELWALL, Mike. **Heart and soul: Sentiment strength detection in the social web with sentistrength**, 2013

[15] WANG, Hao. CAN, Dogan. KAZEMZADEH, Abe. BAR, Fraçois. NARAYANAN, Shrikanth. **A system for real-time Twitter sentiment analysis of 2012 U.S. Presidential Election Cycle.** In: Proceedings of the ACL 2012 System Demonstrations, pp 115–120, Jeju Island, Korea. Association for Computational Linguistics. 2012.

[16] HU, Minqing, BING, Liu. **Mining and summarizing customer reviews**. In: KDD'04, pp 168-177. 2004. Disponível em: http://doi.acm.org/10.1145/1014052.1014073. Acesso em 15/12/2022.

[17] WILSON, Theresa. HOFFMANN, Paul. SOMASUNDARAN, Swapna, KESSLER, Jason, WIEBE, Janyce, CHOI, Yejin, CARDIE, Claire, RILOFF, Ellen, PATWARDHAN, Siddharth. **OpinionFinder: a system for subjectivity analysis**. In: HLT/EMNLP on interactive demonstrations, pp 34-35.2005.

[18] WILSON, Theresa, WIEBE, Janyce, HOFFMANN, Paul. **Recognizing contextual polarity in phrase-level sentiment analysis.** In: Proceedings of the conference on human language technology and empirical methods in natural language processing (HLT '05), pp 347-354. 2005.

[19] NIELSEN, Finn A. **A new ANEW: evaluation of a word list for sentiment analysis in microblogs.** arXiv:1103.2903. 2011.

[20] BRADLEY, Margaret M, LANG, Peter J. **Affective norms for English words (ANEW): stimuli, instruction manual, and affective ratings.** Technical report, Center for Research in Psychophysiology, University of Florida, Gainesville, FL. 1999.

[21] TABOADA, Maite. BROOKE, Julian. TOFILOSKI, Milan. VOLL, Kimberly. STEDE, Manfred**. Lexicon-based methods for sentiment analysis**. Comput Linguist 37(2):267-307. 2011.

[22] HANNAK, Aniko. ANDERSON, Eric. BARRETT, Lisa F. LEHMANN, Sune. MISLOVE, Alan. RIEDEWALD, Mirek. **Tweetin' in the rain: exploring societal-scale effects of weather on mood.** In: 6th international AAAI conference on weblogs and social media (ICWSM). 2012.

[23] MOHAMMAD, Saif. **#emotional tweets. In: The first joint conference on lexical and computational semantics** - volume 1: proceedings of the main conference and the shared task, and volume 2: proceedings of the sixth international workshop on semantic evaluation (SemEval 2012), pp 246-255. 2012. Disponível em: http://www.aclweb.org/anthology/S12-1033. Acesso em 17/10/2022.

[24] MOHAMMAD S, TURNEY PD. **Crowdsourcing a word-emotion association lexicon.** Comput Intell 29(3):436-465. 2013.

[25] PAPPAS, Nikolaos, BELIS-POPESCU, Andrei. **Sentiment Analysis of User Comments for One-Class Collaborative Filtering over TED Talks**. In SIGIR, 2013.

[26] MOHAMMAD SM, KIRITCHENKO S, ZHU X (2013) **NRC-Canada: building the state-of-the-art in sentiment analysis of tweets**. In: Proceedings of the seventh international workshop on semantic evaluation exercises (SemEval 2013).

[27] SOCHER R, PERELYGIN A, WU J, CHUANG J, MANNING CD, NG AY, POTTS C. **Recursive deep models for semantic compositionality over a sentiment treebank**. In: Proceedings of the 2013 conference on empirical methods in natural language processing (EMNLP '13), pp 1631-1642. 2013.

[28] LEVSALLOIS C**. Umigon: sentiment analysis for tweets based on terms lists and heuristics. In: The second joint conference on lexical and computational semantics** (*SEM), volume 2: proceedings of the seventh international workshop on semantic evaluation (SemEval 2013), pp 414-417. 2013. Disponível em http://www.aclweb.org/anthology/S13-2068. Acesso em: 8/1/2023.

[29] HUTTO C. J. GILBERT, Eric. **VADER: A Parsimonious Rule-Based Model for Sentiment Analysis of Social Media Text**. 2014. In: Proceedings of the Eighth International AAAI Conference on Weblogs and Social Media. Pág. 216-225. Disponível em https://doi.org/10.1609/icwsm.v8i1.14550. Acesso em 7/10/2022.

[30] CHEN Taianqi, GUESTRIN Carlos. **XGBoost: A Scalable Tree Boosting System**. 2016. ArXiv:1603.02754v3.

[31] BEKKERMAN, Ron. **The Present and the Future of the KDD Cup Competition: an Outsider's Perspective**. In: LinkedIn. 2015. Disponível em: https://www.linkedin.com/pulse/present-future-kdd-cup-competition-outsiders-ron-bekkerman/. Acesso em: 9/1/2023.

[32] **Tweepy: An east-to-use Pyhton library for accessing the Twitter API**. Disponível em: https://www.tweepy.org/. Acesso em: 10/1/2023.

[33] **Twitter API v2**. Disponível em: https://developer.twitter.com/en/docs/api-reference-index. Acesso em 8/1/2023.

[34] HU, Kelvin, GRIMBERG, Daniella, DURDYEV, Eziz. **Twitter Sentiment Analysis for Predicting Stock Price Movements.**

Aprendizado Federado Aplicado à Classificação de Doenças Pulmonares em Imagens de Raio-X

1st Weld Lucas Cunha
Cientista de dados no SiDi
Campinas, Brasil
weld.c@sidi.org.br

2nd Cesar Christian Castelo Fernandez
Cientista de dados sênior no SiDi
Campinas, Brasil
cesar.f@sidi.org.br

3rd Samuel Botter Martins
Professor no IFSP
Campinas, Brasil
samuel.martins@ifsp.edu.br

***Resumo*—Aprendizado Federado (ou FL, do inglês *Federated Learning*) é uma abordagem de aprendizado de máquina em que muitos clientes (por exemplo, dispositivos móveis ou organizações) são capazes de treinar um modelo matemático, de maneira colaborativa, sob a orquestração de um servidor central, mantendo os dados de treinamento descentralizados e também mantendo os dados dos usuário privados [1]. Neste projeto, uma estrutura de aprendizado federado é desenvolvida para treinar um modelo de aprendizado profundo (DL, do inglês *Deep Learning*) visando identificar doenças torácicas em imagens médicas de diferentes hospitais. Foi observado um grande potencial referente a aplicação do aprendizado federado neste tipo de contexto, tanto por possibilitar que o modelo final tenha um melhor desempenho que os modelos locais, conforme é mostrado no desenvolver deste trabalho, quanto em relação à necessidade de manter a privacidade dos dados.**

***Index Terms*—Aprendizado Federado, Privacidade, Imagens Médicas, Doenças Torácicas**

I. Introdução

Visando garantir que os dados de treinamento permaneçam em dispositivos pessoais e possibilitar que o aprendizado de máquina seja implementado de maneira colaborativa e com modelos complexos entre dispositivos distribuídos, uma abordagem descentralizada de *Machine Learning* (ML) chamada *Federated Learning* é introduzida em [2]. *Federated Learning* é uma abordagem de aprendizado de máquina em que muitos clientes (por exemplo, dispositivos móveis ou organizações) são capazes de treinar um modelo, de maneira colaborativa, sob a orquestração de um servidor central, mantendo os dados de treinamento descentralizados e também mantendo a privacidade dos dados de seus usuários [1].

Motivado pelo crescimento explosivo no uso de dispositivos móveis e pela necessidade de mitigar muitos dos riscos sistêmicos relacionados à privacidade dos dados, essa técnica se tornou uma das chaves para resolver muitos dos problemas atuais da indústria de aprendizado de máquina [3], [4]. No entanto, apesar dos recentes avanços e do crescente interesse na pesquisa sobre o tema, essa área ainda apresenta uma extensa coleção de problemas e desafios abertos.

Um ciclo de trabalho típico do desenvolvimento de um sistema de aprendizado federado é mostrado na Fig. 1 e os estágios principais desse processo são descritos abaixo.

1) Identificação do problema: A equipe de desenvolvimento e os líderes de negócio escolhem um problema a ser resolvido no qual o aprendizado federado possa ser aplicado como solução técnica.
2) Instrumentação do cliente: Se necessário, os clientes devem estar preparados para coletar e armazenar dados localmente, além de executar as próximas etapas do processo, como o treinamento de modelos e enviar os modelos treinados para o servidor central. Em alguns casos, dados ou metadados adicionais podem precisar ser mantidos, por exemplo, dados de interação do usuário para fornecer rótulos para uma tarefa de aprendizado supervisionado.
3) Prototipagem de modelo: A arquitetura do modelo deve ser escolhida para que todos os clientes e o servidor central possam otimizar o modelo inicial. O protótipo de aprendizado de hiperparâmetros pode ser escolhido em uma simulação de FL usando um conjunto de dados proxy e o modelo inicial é distribuído a todos os clientes que participam do processo de treinamento federado.
4) Treinamento do modelo federado: Várias tarefas de treinamento são iniciadas para treinar diferentes variações do modelo ou usar diferentes hiperparâmetros de otimização.
5) Avaliação do modelo: Após a execução das taregas de treinamento (a quantidade de tempo varia dependendo do contexto comercial do modelo), os modelos são avaliados de maneira automatizada, podendo também ser analisados por uma equipe técnica. A análise pode in-

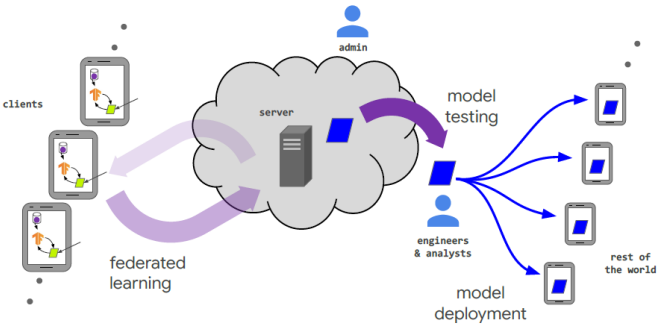


Figura 1. O ciclo de vida de um modelo treinado por FL e os vários atores em um sistema de aprendizado federado.

cluir métricas calculadas nos conjuntos de dados padrão no data center, ou avaliação federada, na qual os modelos são impulsionados para clientes mantidos para avaliação nos dados locais do cliente.

6) Implantação: Finalmente, uma vez que um bom modelo é selecionado, ele é implantado em todos os clientes por meio de um processo de lançamento de modelo padrão. A metodologia FL é um processo iterativo e pode ser repetida muitas vezes.

Nas próximas seções explicaremos em mais detalhe sobre a metodologia do federated learning e os exprimentos conduzidos, assim como os resultados obtidos. Na seção II discutimos sobre o método proposto e o problema abordado neste trabalho. Na seção III, detalhamos os experimentos, dados utilizados, o modelo base e detalhamos o protocolo de experimentos. Na seção IV os resultados são apresentados e discutidos visando um maior entendimento do que os expriemntos nos proporcionaram. Na seção V é realizada a conclusão do trabalho e definição de trabalhos futuros.

II. Método Proposto

A metodologia de aprendizado federado pode ser aplicada a muitos contextos em que hajam clientes descentralizados ou quando a privacidade dos dados é uma grande preocupação, como dados sensíveis oriundos de exames médicos, por exemplo. Muitos hospitais coletam dados de seus pacientes, o que é uma ótima fonte de informação para os modelos de aprendizado de máquina. Compartilhar esses dados com outros hospitais e criar uma grande base de dados seria uma boa maneira de treinar melhores modelos, no entanto, devido a questões relacionadas à privacidade dos pacientes, isso não é possível na grande maioria dos casos. Neste trabalho, aplicamos a metodologia de aprendizado federado a um cenário com vários hospitais, considerando dados de imagens médicas e visando identificação de doenças pulmonares. É importante ressaltar que as imagens não são compartilhadas entre os hospitais, e apenas os modelos treinados individualmente com os dados de cada hospital são enviados a um servidor central. Os principais objetivos deste trabalhos são listados abaixo:

- Desenvolver uma estrutura fácil de usar e replicável para treinamento e implantação de modelos de aprenziado federado;
- Manter a privacidade de dados entre as diferentes fontes de dados (hospitais);
- Obter um modelo final que apresente melhores resultados (medido através de métricas como acurárcia, precisão, f1-score, etc.) do que os modelos treinados individualmente por cada hospital.

O problema a ser resolvido pelo modelo de visão computacional é a identificação de doenças em imagens médicas, ou seja, dada uma imagem de raio-X da região do tórax, nosso modelo deve entender se esta é uma imagem em que há doença ou não (Classificador Binário). Nossa hipótese é que um modelo treinado com aprendizado federado tenha um desempenho melhor do que os modelos treinados individualmente por cada hospital, pois com o aprendizado federado é esperado que o modelo obtenha o conhecimento presente em todas as bases de dados distribuídas.

III. Experimentos

Os experimentos foram conduzidos considerando um cenário com diversos hospitais, nesta seção, os experimentos serão descritos de forma a detalhar sobre os dados utilizados e a configuração utilizada na condução dos experimentos.

A. Dados

Utilizamos o conjunto de dados ChestX-Ray14, lançado por [5]. Este dataset contém 112120 imagens de raios-X de visão frontal do tórax, referentes a 30805 pacientes únicos. Uma amostra com algumas imagens do dataset é mostrada na Fig. 2.

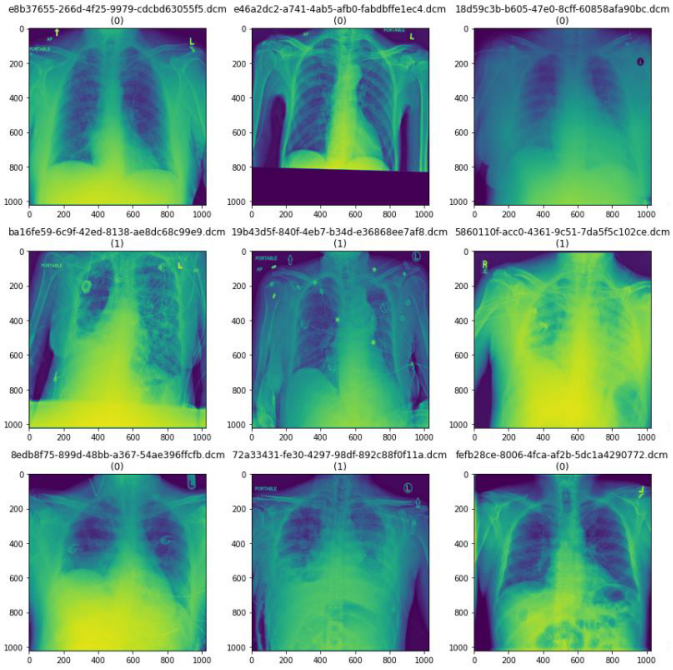


Figura 2. Amostra de imagens da base de dados ChestX-Ray14.

Cada imagem foi anotada com até 14 rótulos de patologias torácicas diferentes usando métodos de extração automática em relatórios de radiologia. No nosso experimento, todas as imagens que apresentavam alguma das patologias foram anotadas como exemplos positivos (em que há doença) e todas as outras imagens que não apresentam nenhuma das 14 possíveis patologias tratadas no dataset foram rotuladas como exemplos negativos (não há doença: saudáveis) para os nossos experimentos.

B. Modelo

O modelo base adotado foi a CheXNet [6], uma rede convolucional densa de 121 camadas. Todo o processamento dos dados e metodologia de treinamento da rede foi definido conforme descrito em [6]. Substituímos a camada final totalmente conectada por uma que possui uma única saída, após

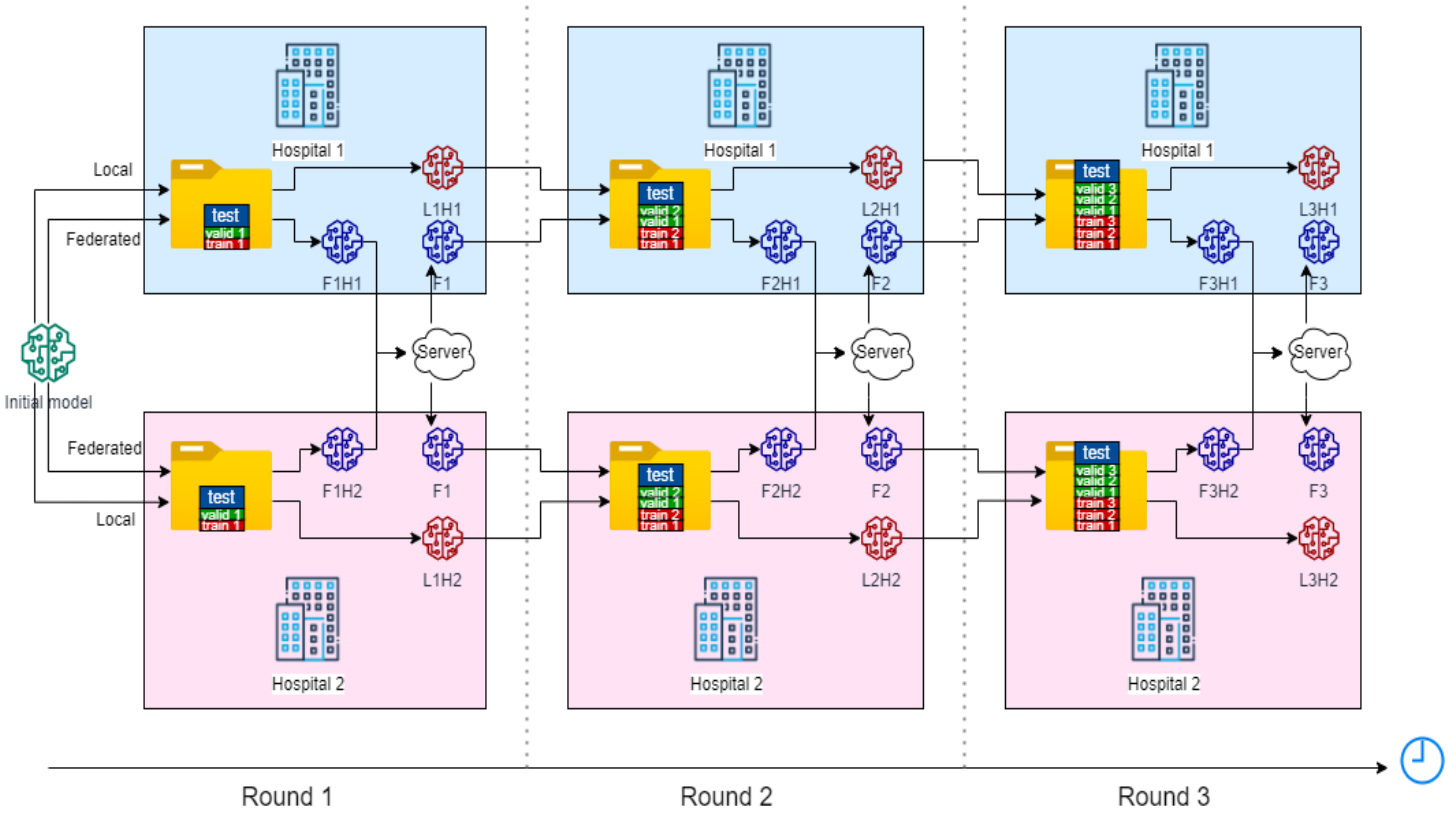


Figura 3. Protocolo de experimentos.

a qual aplicamos uma não linearidade sigmóide. Os pesos da rede são inicializados com pesos de um modelo pré-treinado na base de dados Imagenet [7]. A rede é treinada de ponta a ponta usando o otimizador Adam com parâmetros padrão (β_1 = 0,9 e β_2 = 0,999). Treinamos o modelo usando minibatches de tamanho 16. Utilizamos uma taxa de aprendizado inicial de 0,001 que é deminuída por um fator de 10 a cada vez que a taxa de perda na validação atinge um platô e escolhemos o modelo com a menor perda de validação.

C. Protocolo de Experimentos

O protocolo de Experimentos foi definido e executado conforme os seguintes critérios:

1) Há 2 cenários analisados em paralelo: No primeiro cenário os hospitais participam de um processo de aprendizado federado, contribuindo com seus modelos locais e recebendo um modelo federado a cada iteração. No segundo cenário cada modelo é treinado individualmente por cada hospital, sem qualquer contato com os outros hospitais e considerando apenas os próprios dados.
2) Um modelo inicial é escolhido e treinado anteriormente utilizando os dados da imagenet, o mesmo modelo inicial é usado para o treinamento local e também para as iterações federadas;
3) O conjunto de dados original [5] será dividido em 5 subconjuntos (5 hospitais) com aproximadamente 22 mil imagens cada;
4) Cada subconjunto será dividido em treino, validação e teste;
5) 5 iterações (federadas e locais) serão executadas para cada hospital com 40%/55%/70%/85%/100% dos dados disponíveis (acumulativos).
6) Um total de 5 x 5 x 2 = 50 rotinas de treinamento são executadas, sendo 25 para as iterações federadas e 25 para os treinamentos locais.

O protocolo de experimentos é resumido na Fig. 3, considerando uma versão mais simples do problema (com 2 hospitais e 3 iterações), onde os modelos locais são representados em vermelho e os modelos federados em azul. Podemos observar que o mesmo modelo inicial é adotado para os 2 cenários, há também uma atualização das bases de dados de treino e validação a cada iteração/novo treinamento, porém a base de testes é mantida inalterada ao longo de todo o processo. Posteriormente, cada modelo local terá sua performance comparada ao modelo federado a cada iteração, considerando cada uma das bases locais. A nomenclatura reservada aos modelos na Fig. 3 é apresentada a seguir:

- L1H1: Modelo local da iteração de treinamento 1 do hospital 1.
- F1H1: Modelo federado intermediário (após treinamento local) da iteração de treinamento 1 do hospital 1.
- F1: Modelo federado após execução do processo de

junção pelo servidor central, da iteração de treinamento 1 do hospital 1. Este é o modelo que resulta após 1 iteração federativa completa.

IV. Resultados e Discussão

Após a execução dos experimentos, foi possível observar a performance dos modelos ao longo das iterações de treinamento local e federado, A acurácia de cada modelo na base de testes ao longo das 5 iterações é mostrada na Fig. 4.

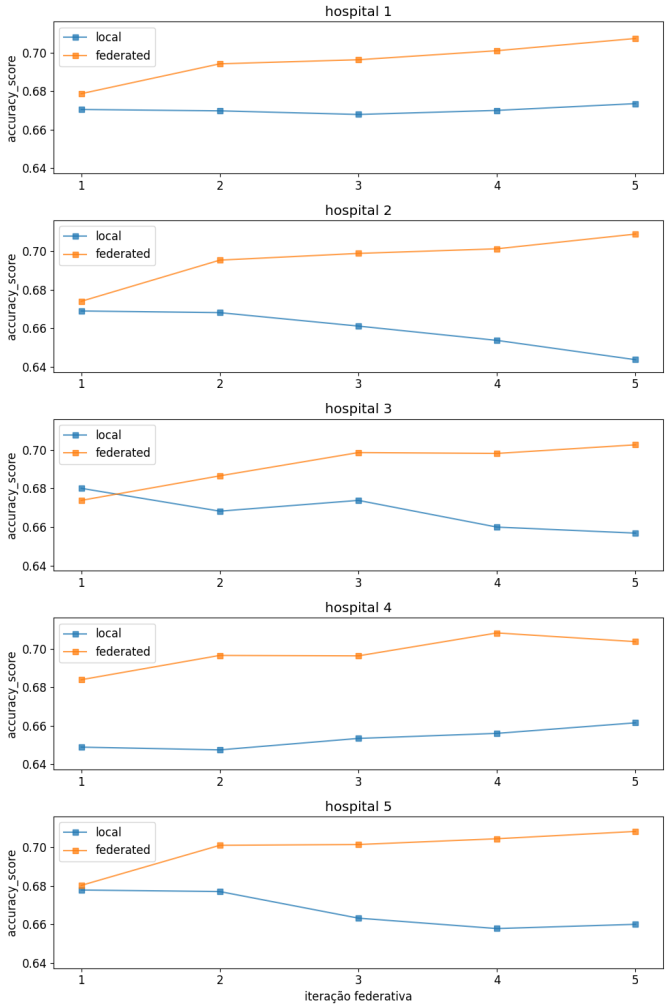


Figura 4. Modelos locais vs modelo federado: comparação da performance ao longo das iterações de treinamento.

Para cada hospital, é realizada a análise comparativa entre o seu modelo treinado localmente e o modelo federado. Considerando que a cada rodada temos 1 modelo federado para o conjunto de hospitais, e n modelos locais (onde n é o número de hospitais), teremos portanto $n = 5$ comparações entre os modelos locais e o modelo federado ao longo das 5 rodadas. Na Fig. 4 fica clara a vantagem do modelo federado em relação aos modelos treinados localmente, é nítida a diferença da acurácia entre os modelos, sendo que o modelo federado tem melhor acurácia ao longo da maioria das iterações e considerando todas as bases locais. Em média, considerando todos os hospitais e todas as iterações, a acurácia aumentou $3,24$ pontos percentuais com o modelo federado em relação aos modelos locais.

É interessante notar também que, mesmo quando os modelos locais pioram seus resultados ao longo das iterações, o modelo federado permaneceu apresentando evolução de suas métricas, iteração a iteração. Chegando a atingir um patamar de 70% de acurácia, enquanto os modelos locais chegam a valores próximos de 67%.

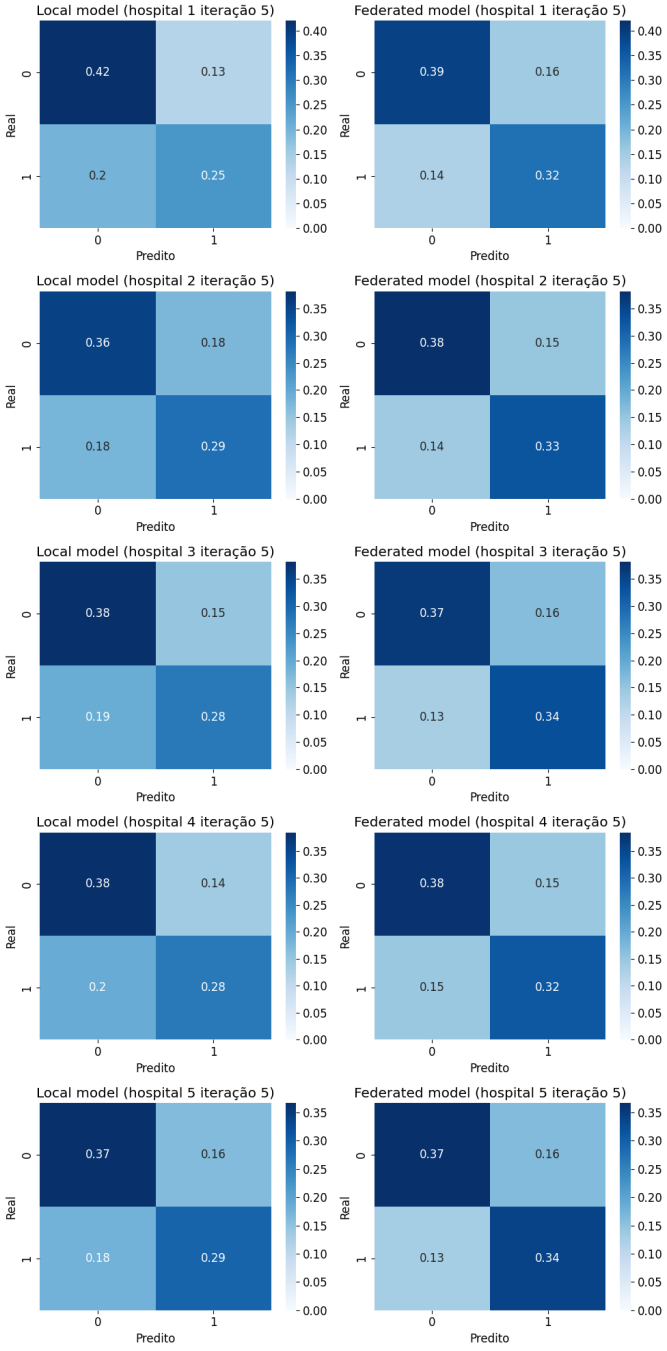


Figura 5. Matrizes de confusão após treinamento.

Site number	Accuracy	Balanced Accuracy	Recall	Precision	F1-Score	ROC AUC
1	2,52%	2,61%	3,59%	2,78%	3,13%	2,61%
2	3,65%	3,58%	2,47%	4,65%	3,47%	3,58%
3	2,42%	2,27%	-0,02%	3,40%	1,70%	2,27%
4	4,42%	4,59%	7,65%	4,25%	6,14%	4,59%
5	3,19%	3,26%	4,38%	3,30%	3,73%	3,26%

Tabela I
DIFERENÇA MÉDIA EM PONTOS PERCENTUAIS ENTRE OS MODELOS LOCAIS E O MODELO FEDERADO

A Tabela I apresenta um compilado com o valor de aumento médio em pontos percentuais ao longo das 5 iterações considerando diferentes métricas para cada um dos locais (ou *site*, em inglês). Observa-se que, em geral, todos os locais apresentaram melhoria das métricas se comparando cada modelo local ao seu par federado. Há um único exemplo em que houve piora do valor médio, por um valor relativamente baixo, enquando que na grande maioria houve um aumento considerável, chegando a mais de 7 pontos percentuais em alguns casos.

Na Fig. 5 é apresentada a comparação entre a as matrizes de confusão do modelo local e modelo federado, para cada um dos hospitais, após todas as rodadas de treinamento. Pode-se observar que na grande maioria dos casos, o modelo federado obteve melhor performance, tendo mais predições corretas tanto da classe negativa (imagens saudáveis) quanto positiva (imagens de tórax com alguma doença). Enquanto nos modelos locais, há um leve desequilíbrio em alguns casos, havendo mais falso-positivos do que falso-negativos, quando se trata dos erros cometidos pelos modelos. Por outro lado, pode-se observar que há um maior equilíbrio entre os falso-positivos e falso-negativos nos modelo federado.

V. CONCLUSÃO

O aprendizado federado oferece uma ótima oportunidade para manter a privacidade dos dados dos clientes, especialmente quando se trata de clientes com dados sensíveis. Também é possível melhorar o desempenho do modelo de forma a promover o benefício mútuo para todos os dispositivos ou instituições envolvidas.

É importante destacar que algumas suposições precisam ser seguidas para esperar bons resultados desta técnica. As distribuições de dados devem ser semelhantes entre as diferentes fontes de dados e a padronização dos dados e processos é uma obrigação, visando manter a coerência e possibilitar que o modelo federado aprenda de maneira adequada. Foi desenvolvido uma estrutura local que possibilitou a execução do treinamentos dos modelos federados e locais, possibilitando uma posterios comparção entre estes dois contextos.

Como trabalho futuro, pretendemos expandir o modelo para um classificador multiclasse e multilabel, tratando as 14 patologias presentes no dataset. Também pretendemos utilizar outras configurações, visando melhorar o treinamento do modelo, como alteração do otimizador e outras técnicas que possam melhorar a performance do nosso modelo em geral. A utilizaçao de um framework que possibilite a execução do treinamento em uma núvem pública também faz parte da sequência de desenvolvimento desta linha de pesquisa, possibilitando a execução das rotinas de treinamento de maneira remota e com participação de diversas fontes de dados.

REFERÊNCIAS

[1] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings, *et al.*, "Advances and open problems in federated learning," *Foundations and Trends® in Machine Learning*, vol. 14, no. 1–2, pp. 1–210, 2021.

[2] H. B. McMahan, E. Moore, D. Ramage, and B. A. y Arcas, "Federated learning of deep networks using model averaging," *arXiv preprint arXiv:1602.05629*, vol. 2, 2016.

[3] W. Y. B. Lim, N. C. Luong, D. T. Hoang, Y. Jiao, Y.-C. Liang, Q. Yang, D. Niyato, and C. Miao, "Federated learning in mobile edge networks: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 2031–2063, 2020.

[4] T. Ryffel, A. Trask, M. Dahl, B. Wagner, J. Mancuso, D. Rueckert, and J. Passerat-Palmbach, "A generic framework for privacy preserving deep learning," *arXiv preprint arXiv:1811.04017*, 2018.

[5] X. Wang, Y. Peng, L. Lu, Z. Lu, M. Bagheri, and R. M. Summers, "Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2097–2106, 2017.

[6] P. Rajpurkar, J. Irvin, K. Zhu, B. Yang, H. Mehta, T. Duan, D. Ding, A. Bagul, C. Langlotz, K. Shpanskaya, *et al.*, "Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning," *arXiv preprint arXiv:1711.05225*, 2017.

[7] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255, Ieee, 2009.

Transfer Learning for Personalization in Federated Learning on Edge Devices

Alexandre Freire S. Osorio
Instituto Federal de São Paulo
Campinas, Brazil
alexandreosorio@yahoo.com

Samuel Botter Martins
Instituto Federal de São Paulo
Campinas, Brazil
samuel.martins@ifsp.edu.br

Francisco Ubaldo Vieira Jr.
Instituto Federal de São Paulo
Campinas, Brazil
ubaldo@ifsp.edu.br

***Abstract*—Federated learning (FL) is a promising technique to cope with privacy and legalization issues in distributed learning scenarios. Also, FL has the potential of increasing the generalizability and personalizability of models. At the same time, HAR (Human Activity Recognition) is gaining momentum due to advances in IoT, sensors and AI. Training machine learning models on Android edge devices in a federated way brings the core processing closer to the data source, while optimizing the network resources. In such an environment, the heterogeneity of the computational capacity of the devices is a challenge, and the complexity of the models is a sensitive issue. The present work proposes a method to personalize the models with the local data of each client in an FL scenario where the training is performed on Android smartphones. A Multilayer Perceptron (MLP), fine-tuned with local data, is used to recognize human activities from sensors' data extracted from the Extrasensory dataset. A base model, pre-trained with a large amount of data, works as a feature extractor for the head model, which is effectively federated-trained on the devices. The results of the method tested in a small FL configuration with four smartphones are presented.**

***Index Terms*—federated learning, model personalization, human activity recognition, transfer learning, edge AI**

I. Introduction

The key idea of Federated Learning (FL) is to provide data privacy in a distributed learning scenario, where it is possible to share knowledge acquired over the data of the clients (the users in FL) [1]. Training is performed locally on the client using its local data which are not shared across the network, but only the coefficients of the trained model. In general, the FL operation is composed of the following steps [2]: (a) the FL server determines the model to be trained on the clients; (b) a subset of clients is chosen; (c) the server shares the global model to the selected clients; (d) the clients train the model locally; (e) each client sends updates to the server. After the last step, the server aggregates the parameters of each client to update the global model, and then the flow returns to step (c) to start a new round. Differently from traditional machine learning, transfer learning aims to work in situations where the domain, tasks and/or data distribution are different from training to inference/test conditions [3]. The idea of transferring the knowledge acquired from a previous task to a target task can be used to personalize the model with the target data. Scenarios where the data distribution across clients in FL are non-IID (independent and identically distributed) require model personalization [4]. Model personalization is therefore necessary when dealing with a dataset as diverse as Extrasensory, in which the data was acquired in-the-wild, making the class distribution among individuals very diverse. Extrasensory is formed by data from the sensors of smartphones and smartwatches of 60 participants.

Due to the growing availability of embedded sensors, as well as advancements in ML, AI, and IoT, the research topic of Human Activity Recognition (HAR) is getting more attention in recent years. Techniques of HAR from sensors' data can be applied in areas like health [5]. This work presents a method for model personalization in FL on Android devices, based on transfer learning. Edge devices have restricted computing resources. As a tentative to reduce the complexity of the models to be trained on the devices, the method considers the use of Multilayer Perceptron (MLP) instead of more complex structures, like deep learning models. The training on the device is implemented as a fine-tuning of a base model previously trained with a large dataset. An experiment applying the method to HAR from sensors' data extracted from the Extrasensory dataset [6], in a small FL setup with four Android smartphones as the clients, was conducted. Results showing the model structures and the metrics obtained in an FL experiment with four Android devices are presented.

II. Method

The model personalization proposed consists of two phases: offline and Federate Learning. The offline phase starts by building an MLP centrally trained on a PC with a large and diverse dataset, called base model. The optimum hyperparameters are searched using Keras Hypertuning. The bottleneck model is obtained by freezing the coefficients and removing the classification layer from the trained base model. A second, thin, head model is the one that will be federated-trained on Android smartphones. Both models are serialized in Tensorflow Lite format and saved to the device memory.

The experiment used the Extrasensory as the source of train, validation and test data, in both phases. To configure the FL phase, each smartphone loads the data of one of the participants of the dataset that was not used to train the base model. During the training, the bottleneck features extracted from these data serve as the input of the head model.

A. *Extrasensory dataset*

Extrasensory is a dataset of sensors data that was built with 60 participants, who performed a range of physical activities and daily activities (e.g. sleeping, watching TV, walking, running, etc.), in different locations (e.g. school, home, work) [6]. The data were collected from sensors like accelerometer, gyroscope, magnetometer, clock accelerometer, location, and audio, of smartphones and smartwaches devices. The complete dataset has more than 300,000 minutes of collection. The data were captured "in the wild", which means that each participant was free to define the way they use the devices during their day-by-day. The dataset is multi-label, since participants were free to associate more than one label for a given sample. To simplify the HAR on Extrasensory, this work considered only the set of classes (the human activities) named primary, which contains the classes that are mutually exclusive. In this way, the classification problem becomes multi-class, instead of multi-label. The set of primary classes is: *standing*, *sitting*, *lying down*, *running*, *walking* and *bicycling*. The main characteristics of Extrasensory are listed below:

- 60 participants
- Over 300k samples
- Sensors:
  high-frequency motion-reactive: accelerometer, gyroscope, magnetometer
  location services
  audio
  watch compass
  phone status: app status, battery state, Wi-Fi availability, on the phone, time-of-day
- Pseudo-sensors (processed):
  calibrated version of gyroscope (tries to remove drift effect)
  unbiased version of magnetometer (tries to remove bias of the magnetic field created by the phone itself)
  gravitation direction (magnitude is always 1G)
  user-generated acceleration (raw acceleration minus gravitation acceleration)
  estimated orientation of the phone
  rotation vector
- In-the-Wild: data was collected from users that were engaged in their regular natural behavior
- Labels:
  Main activity (mutually exclusive): lying down, sitting, standing in place, standing and moving, walking, running, bicycling
  Secondary activity. Additional 109 labels describing more specific context in different aspects
  Multiple secondary labels can be applied to an example

B. *Steps of the method*

The two phases are summarized below. Fig. 1 shows a representation of the two phases.

- Offline phase: define the structures of the base model and the head model, train the base model and convert both in TFLite format. The code, written in Python, runs on a GPU of a machine-learning server
- FL phase: a Java code runs on the Android devices, which communicate with the FL server running on a PC and written in Python

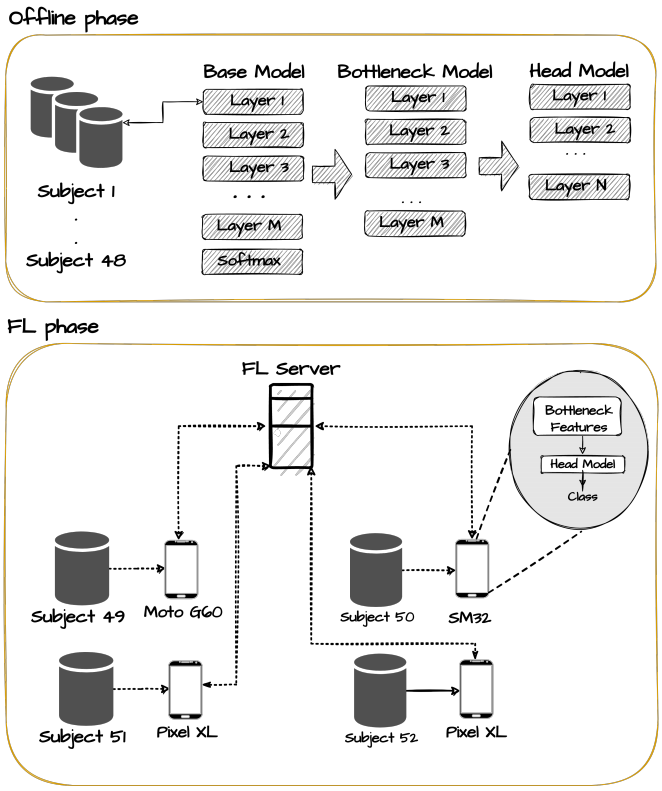


Fig. 1. The two phases of the proposed method.

For the offline phase, the steps were the following:

1) Randomly choose a set of 48 participants of the Extrasensory dataset and concatenate their data in a single csv file
2) Split the csv file into train, validation and test data, randomly picking samples in a ratio of 75/15/10 (%)
3) Train the base model with the train data, using the Keras Hypertuning to search the hyperparameters that maximize the balanced accuracy
4) Freeze the parameters of the base model and remove the classification layer
5) Convert the bottleneck and the head models to serialized Tensorflow Lite (.tflite) format
6) Copy the .tflite files to the appropriate Android project model directory
7) Split the data of each participant of the dataset into train and test data and copy the csv files to the Android project data directory

For the FL phase, the steps are the following:

1) For each of the four smartphones, select a different participant of Extrasensory, whose data were not used to train the base model;

2) Configure the hyperparameters of the FL and start the train on each device.

III. Results and Discussion

The metric adopted was the balanced accuracy, since precision and F1 score are very sensitive in cases of rare labels [6], as is the case of Extrasensory. Equation (1) represents the balanced accuracy formula. The categorical cross-entropy was used as the loss function for the base model training, as it is more appropriate to one-hot encoded label vectors. Fig. 2 shows the distribution of the primary classes for the set of joined 48 participants used to train the base model. The axis representing the number of samples is on a logarithmic scale, due to the disproportion between the classes. In general, there are many more samples of non-movement activities than movement.

$$BA = \frac{(sensitivity + specificity)}{2} \quad (1)$$

A. Training the base model with a subset of the features

Extrasensory has a large amount of 225 features, corresponding to raw data from sensors as well as statistics calculated on the raw data. In a real-time situation, it can be hard to obtain all these features due to the computational effort needed to calculate the statistics and because we cannot guarantee that the device is equipped with the complete set of sensors used in Extrasensory. The first approach was to try to build an MLP that could achieve a good balanced accuracy if trained with a subset of the features. The subset chosen comprises 48 features from the raw data of three sensors – gyroscope, accelerometer and magnetometer – and their statistics: mean, standard deviation, moment 3, moment 4, percentile 25, percentile 50 and percentile 75.

The balanced accuracy obtained was 0.7701, and 0.8080 for the categorical cross-entropy loss, for the validation data. It was configured 200 epochs. Fig. 3 shows the results for the training and validation data.

B. Training the base model with the complete set of features

Training the base model with the complete set of 225 features of the dataset resulted in better metrics: 0.8954 for the balanced accuracy and 0.3044 for the categorical cross-entropy loss. The hypothesis to explain the difference between the two results is that the partial set of features comprises only sensors that are sensitive to movement (accelerometer, gyroscope and magnetometer) and, therefore, the model has low competence to separate between sitting, standing and lying down, the three activities in which the participant was in non-movement situations.

As already explained, both the base model and the head model had their best hyperparameters calculated using Keras Hypertuning, using Bayesian optimization. The defined search space for each model is listed below.

Search space for the base model:

- learning rate: 0.01 or 0.0001
- dropout rate: 0.15 or 0.05
- number of internal layers: 2 to 4
- activation function: relu or tanh

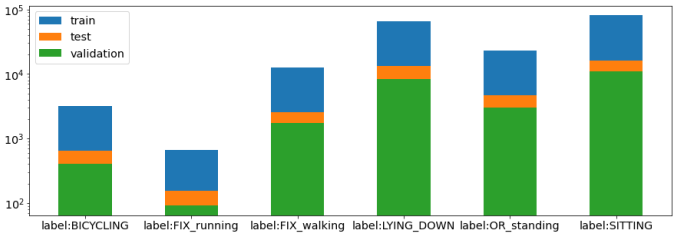


Fig. 2. Distribution of samples per class for the set of 48 participants.

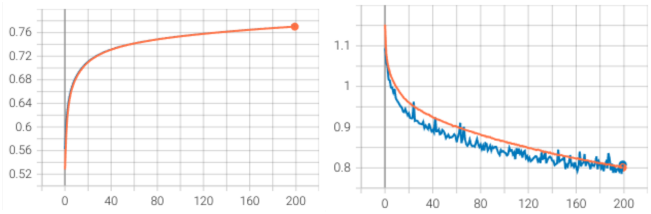


Fig. 3. Balanced accuracy (left) and loss (right) for the train (red) and validation (blue) data, for the base model, using the partial set of features. It was configured 200 epochs and a batch size of 32, as well as early stopping.

Search space for the head model:

- learning rate: 0.0001
- dropout rate: 0.15 or 0.05
- number of internal layers: 1 to 4
- activation function: relu or tanh

The resulting optimal hyperparameters are listed below. It is worth noting that the resulting head model is very light, which is a valuable characteristic when it comes to edge AI, especially when the training is on the device. Fig. 4 represents the resulting MLP structure for both models.

Base model

- learning rate: 0.0001
- dropout rate: 0.05
- number of internal layers: 2
- activation function: relu
- number of neurons for the 1st layer: 288
- number of neurons for the 2nd layer: 128

Head model

- learning rate: 0.0001
- dropout rate: 0.05
- number of internal layers: 1
- activation function: tanh
- number of neurons of the layer: 24
- number of training coefficients: 3,246

The training of the base model was configured with 200 epochs, batch size of 32 samples, as well as early stopping to monitor the loss with patience argument of 50 samples. Fig. 5 shows the curves for the balanced accuracy and the loss for train and validation data for the base model.

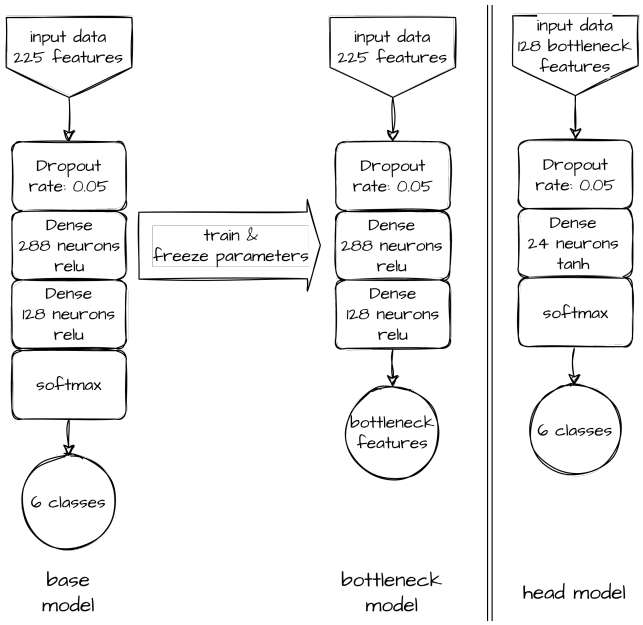


Fig. 4. Base model and head model, as built during the offline phase.

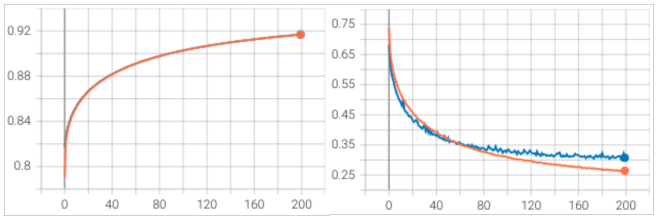


Fig. 5. Balanced accuracy (left) and loss (right) for the train (red) and validation (blue) data, for the base model, using the full set of features. It was configured 200 epochs and a batch size of 32, as well as early stopping with patience argument of 50 samples.

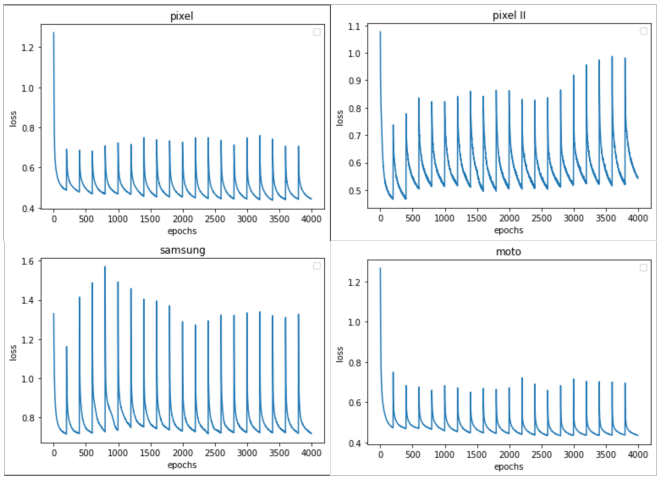


Fig. 6. Loss along the 20 rounds, with 200 epochs per round, for each client.

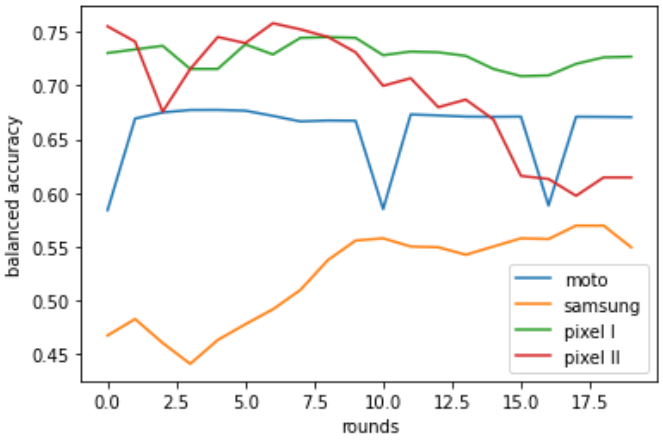


Fig. 7. Test balanced accuracy at the end of each of the 20 rounds.

C. Federated Learning on Android Devices

The FL experiment run on 4 clients (smartphones) of the following models:

- Motorola G60 (physical device)
- Samsung Galaxy SM-M325FV (physical device)
- Pixel XL API 30 (emulated device)
- Pixel XL API 30 (emulated device)

Each device simulates de acquisition of its own sensors' data by reading the data of one of the participants of the Extrasensory dataset. These four participants must be outside the set of 48 participants whose data were used to train the base model. The train data is shuffled on the device before choosing the samples to compose each batch of data.

The FL server is a Python code that runs on a PC whose IP number and access port were previously configured in each one of the four devices. The model aggregation algorithm used was FedAvg, which is the most common aggregation algorithm in use [7]. Regarding the FL server configuration, both the fraction of clients to fit and the fraction of clients to evaluate was set to 100

The FL on the smartphones resulted in low balanced accuracies. Fig. 6 shows the categorical cross-entropy loss for each client, along the rounds, while Fig. 7 shows the balanced accuracy for the test set, calculated at the end of each round.

An important future work would consider 12 clients which is the number of participants of the dataset that were not used to train the base model, to check whether the increase in data diversity would result in more accurate models.

Also, it has been proved that time correlations are critical to the performance of the activity recognition model [8]. It is intuitive to think that the time information between samples is an important aspect to consider when extracting knowledge from sensors' data. Another possible future work would build HAR models that consider the sensors' data as time series.

Conclusions and Future Works

This work presented the results of an experiment employing a proposed method to personalize models to recognize human activities from sensors' data of Android smartphones, using

Federated Learning. The method is divided in two phases: offline – which trains a base model from a large amount of data from different participants of the Extrasensory dataset, and defines the head model, for which the base model acts as a feature extractor – and Federated Learning – which trains the head model on each device, in a federated way. During the offline phase of the experiment, the balanced accuracy of the base model was significantly higher when using the full set of 225 features to train the model than a partial set of 48 features (0.7701 to 0.8954).

The offline phase resulted in a very light head model, with only 3,246 trainable parameters, which is an important feature when it comes to edge device training. However, the FL on the four smartphones resulted in low balanced accuracies, considering 20 rounds and 200 epochs per round. A possible next experiment would be to run with a larger set of clients to see if this would introduce a greater enough diversity of label distribution to increase the model's personalization. Such an experiment is most feasible using a device farm cloud service.

Another possible future work would be to build HAR models that consider sensor data as time series. A step to improve the analysis of the results is to implement a device profile analysis, to show the CPU and memory usage curves of the devices throughout the experiment.

ACKNOWLEDGMENT

Acknowledgments for the Instituto Federal de São Paulo for the support of this research. A. F S. O. thanks the support of the Hub for Artificial Intelligence and Cognitive Architectures (H.IAAC – Hub de Inteligência Artificial e Arquiteturas Cognitivas), a project founded by PPI-Softex/MCTI by grant 01245.013778/2020-21 through the Brazilian Federal Government.

REFERENCES

[1] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. "Federated machine learning: Concept and applications," ACM Transactions on Intelligent Systems and Technology (TIST), 10(2):12, 2019.

[2] Abreha, Haftay Gebreslasie, Mohammad Hayajneh, and Mohamed Adel Serhani. "Federated learning in edge computing: a systematic survey," Sensors 22.2 (2022): 450.

[3] S. J. Pan and Q. Yang, "A survey on transfer learning, "IEEE TKDE, vol. 22, no. 10, pp. 1345–1359, (2010).

[4] Han, D. J., Kim, D. Y., Choi, M., Brinton, C. G., and Moon, J. (2022). "SplitGP: Achieving Both Generalization and Personalization in Federated Learning," arXiv preprint arXiv:2212.08343.

[5] Straczkiewicz, M., James, P. and Onnela, JP. "A systematic review of smartphone-based human activity recognition methods for health research," npj Digit. Med. 4, 148 (2021). https://doi.org/10.1038/s41746-021-00514-4

[6] Yonatan Vaizman, Katherine Ellis, and Gert Lanckriet. "Recognizing detailed human context in the wild from smartphones and smartwatches," IEEE pervasive computing, 16(4):62–74, (2017).

[7] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. "Federated learning: Challenges, methods, and future directions," IEEE Signal Processing Magazine, 37(3):50–60 (2020).

[8] Shen, Q., Feng, H., Song, R., Teso, S., Giunchiglia, F., and Xu, H. "Federated Multi-Task Attention for Cross-Individual Human Activity Recognition," Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence (IJCAI-22) (2022). https://www.ijcai.org/proceedings/2022/0475.pdf

INSTITUTO FEDERAL
São Paulo
Câmpus Campinas

CDIF 2023

1º SEMINÁRIO DE CIÊNCIA DE DADOS DO IFSP

ISBN: 978-65-995529-1-5
CBL
9 786599 552915