Detecção de Fake News: Uma Análise Comparativa

Josenildo V. Araújo[1], Paulo Freitas de Araújo Filho[1]

[1]Centro de Informática – Universidade Federal de Pernambuco (UFPE)
Caixa Postal 7851 – 50732-970 – Recife – PE – Brasil

{jva,pfreitas}@cin.ufpe.br

Abstract. *This research addresses Fake News detection through machine learning, emphasizing the effectiveness of the SVM algorithm in conjunction with the TF-IDF feature extractor. The results unveil challenges in handling complex datasets like Liar, underscoring the need for careful choices. The study highlights future opportunities, including in-depth analyses of specific datasets and the integration of advanced Natural Language Processing techniques.*

Resumo. *Esta pesquisa aborda a detecção de Fake News por meio de aprendizado de máquina, destacando a eficácia do algoritmo SVM em conjunto com o extrator de características TF-IDF. Os resultados revelam desafios ao lidar com conjuntos de dados complexos, como o Liar, sublinhando a necessidade de escolhas criteriosas. O estudo aponta oportunidades futuras, incluindo análises mais aprofundadas de conjuntos específicos e a incorporação de técnicas avançadas de Processamento de Linguagem Natural.*

1. Introdução

As redes sociais estão ganhando cada vez mais usuários, conforme o relatório trimestral sobre a internet realizado pela organização especializada em usos digitais, Kepios, mais de 60% da população mundial já está ativa nas redes sociais [1]. O uso cada vez maior destas redes tem levado as pessoas a utilizarem essas mídias como fonte de notícias, de acordo com o relatório Digital News Report 2021 da Reuters Institute for the Study of Journalism, 36% dos entrevistados em todo mundo utilizam as redes sociais como fonte de notícias [2]. Porém nas redes sociais existe muita disseminação de notícias falsas, chamadas de Fake News.

As Fake News são notícias falsas ou distorcidas que circulam na internet e em outros meios de comunicação e o seu objetivo é de enganar, manipular ou influenciar a opinião pública sobre determinados assuntos. Segundo cientistas do Instituto de Tecnologia de Massachusetts – MIT, as Fake News se espalham de maneira 70% mais rápidas do que as notícias verdadeiras [3]. Elas se espalham mais rápido do que as notícias verdadeiras porque exploram as emoções, os preconceitos e os interesses dos leitores, que tendem a compartilhar aquilo que confirma suas crenças ou que causa indignação, medo ou surpresa. Além disso, as Fake News contam com o apoio de robôs, algoritmos e grupos organizados que amplificam sua difusão na internet. Já os impactos gerados destas notícias enganosas são diversos [5], e entre eles estão:

- Prejuízo Financeiro e Emocional: As pessoas podem ser vítimas de golpes, fraudes, extorsões ou chantagens baseadas em informações falsas. Isso resulta em perdas financeiras e sofrimento emocional para os afetados.
- Influência nas Decisões: As Fake News podem distorcer a percepção da realidade e levar a decisões equivocadas. Políticos, empresas e indivíduos podem tomar medidas com base em informações errôneas, afetando suas vidas e a sociedade como um todo.
- Descredibilização dos Meios de Comunicação e Autoridades: A disseminação de notícias falsas minimiza a confiança nas instituições de mídia e nas autoridades. Isso prejudica a credibilidade das fontes de informação legítimas.

Por isso, é importante verificar a veracidade das informações antes de repassá-las e buscar fontes confiáveis e plurais de informação.

Um dos desafios atuais das sociedade é justamente ter mecanismos eficientes de verificação das notícias e classificá-las entre falsas ou verdadeiras para que se possa prevenir a rápida propagação das Fake News. Por essa disseminação ter efeitos devastadores na sociedade, é preciso ter mecanismos que identifiquem as Fake News de forma precisa e confiável.

Este trabalho busca apresentar a reprodução de uma parte de um artigo [4] que propõe uma taxonomia atualizada para modelos de detecção de Fake News baseados em texto e métodos de extração de características, e, realiza um estudo comparativo avaliando várias técnicas de representação de características e abordagens de classificação com base na precisão e no custo computacional.

2. Metodologia

2.1 Revisão da Literatura

Existem vários estudos recentes de classificação de Fake News que procuram utilizar diferentes algoritmos. Abdulrahman e Baykara [15] aplicaram 4 métodos para extrair as características de textos (TF-IDF, vetorização de contagem, vetorização em nível de N-gram e Vetorização em nível de caractere) e 10 classificadores diferentes de aprendizado de máquina e aprendizado profundo (Random forest, KNN, LSVM, Regressão Logística, Naive Bayes multinomial, AdaBoost, XGBoost, ANN, RNN+LSTM, CNN+LSTM) utilizou um conjunto de dados compilado pela comunidade de ciência de dados do Kaggle com 7.796 linhas. Na classificação [15] obteve uma faixa de precisão entre 81 a 100% usando diferentes classificadores, porém por ter utilizado apenas um conjunto de dados pode limitar a generalização e confiabilidade dos resultados.

Já Ozbay e Alatas [14] implementaram 23 algoritmos de classificação supervisionados, nomeados; BayesNet, JRip, OneR, Decision Stump, ZeroR, Stochastic Gradient Descent (SGD), CV Parameter Selection (CVPS), Randomizable Filtered Classifier (RFC), Logistic Model Tree (LMT), Locally Weighted Learning (LWL), Classification Via Clustering (CvC), Weighted Instances Handler Wrapper (WIHW),

Ridor, Multi-Layer Perceptron (MLP), Ordinal Learning Model (OLM), Simple Cart, Attribute Selected Classifier (ASC), J48, Sequential Minimal Optimization (SMO), Bagging, Decision Tree, IBk, e Kernel Logistic Regression (KLR); utilizando o método de extração de características Term Frequency (TF) em 3 conjuntos de dados (BuzzFeed Political News, Random Political News e ISOT). Os resultados da pesquisa [14] destacam a eficácia do algoritmo da Árvore de Decisão na acurácia, precisão e medida F, em comparação aos outros classificadores, enquanto os outros algoritmos funcionam bem no Recall. A acurácia dos classificadores se mostrou baixa no geral em todos os conjuntos de dados, com uma única exceção sendo a Árvore de Decisão no conjunto de dados ISOT apresentando uma acurácia de 0.97, enquanto todas as métricas de acurácia em todos os outros algoritmos de todos os conjuntos de dados ficam entre 0.50 e 0.75.

E por último, Cavalcanti, Cruz e Farhangian [4] realizaram um extenso estudo empírico, implementando 15 técnicas de extração de características (Bag-of-Words (TF), TF-IDF, Word2Vec, GloVe, FastText, ELMo, BERT, RoBERTa, DistilBERT, ALBERT, BART, ELECTRA, XL.Net, LLaMA e Falcon) e 20 algoritmos de classificação (KNN, SVM, Naive Beyes, Logistic Regression, MLP, Random Forest, AdaBoost, XGBoost, CNN, LSTM, ELMo, BERT, RoBERTa, DistilBERT, ALBERT, BART, ELECTRA, XL.Net, LLaMA e Falcon) em 4 conjuntos de dados (Liar, ISOT, George McIntire e COVID). Os resultados da pesquisa [4] mostram que as técnicas ideais de extração de características variam dependendo das características do conjunto de dados, e enfatiza a importância de combinar métodos de representação de características e algoritmos de classificação para obter um melhor desempenho de generalização e mantendo um custo computacional relativamente baixo. Um destaque para o conjunto de dados Liar, que apresentou falta de precisão entre os diferentes classificadores, o que mostra a dificuldade dos classificadores em lidar com conjuntos de dados com rotulações não binárias.

Esta pesquisa se baseia na do artigo “Fake News detection: Taxonomy and comparative study” [4] reproduzindo a taxonomia proposta porém implementando 2 técnicas de extração de características (Bag-of-Words, TF-IDF) e 6 algoritmos de classificação (SVM, Regressão Logística, KNN, Naive Bayes, MLP e Random Forest) em 3 conjuntos de dados (Liar, George McIntire e COVID).

2.2 Técnicas de Extração de Características

A extração de características é um processo fundamental na classificação de textos, pois converte dados de texto em vetores numéricos que podem ser processados por uma máquina. Enquanto a pesquisa de referência [4] implementou uma abordagem abrangente com a aplicação de 15 técnicas distintas, incluindo TF, TF-IDF, Word2Vec, GloVe, fastText, BERT, DistilBERT, ALBERT, RoBERTa, BART, ELECTRA, XLNet, Falcon, ELMO e LLaMA, esta reprodução adotou uma estratégia mais focada limitando-se a duas técnicas, baseadas em contagem, TF (Bag-of-Words) e TF-IDF. Essa simplificação visou otimizar os recursos computacionais com redução de processamento de dados sem comprometer a integridade da análise.

A técnica de Frequência de Termos (TF) [9], aplicada por meio do método Bag-of-Words (BoW), é uma abordagem essencial de extração de características no

processamento de linguagem natural. Essa técnica simplifica documentos de texto complexos ao converter cada documento em um vetor, no qual cada palavra única é atribuída a uma posição específica e sua frequência de ocorrência é quantificada. O modelo Bag-of-Words ignora a ordem das palavras e foca apenas na frequência delas, proporcionando uma representação numérica eficiente para análise de texto. Embora perca nuances semânticas e relacionamentos contextuais, o TF (BoW) é amplamente utilizado em tarefas de classificação de textos, oferecendo simplicidade e eficiência na representação de documentos extensos, como uma estratégia valiosa para a detecção de notícias falsas.

O TF-IDF (Term Frequency-Inverse Document Frequency) [9] é uma técnica avançada de extração de características comumente utilizada no processamento de linguagem natural para análise de texto. Essa abordagem considera não apenas a frequência de uma palavra em um documento específico (TF), mas também sua importância em relação ao conjunto de documentos completo (IDF). O TF-IDF atribui pontuações mais altas a palavras que são frequentes em um documento específico, mas raras em todo o conjunto de dados, destacando termos distintivos e informativos. Ao ponderar a importância relativa das palavras, o TF-IDF proporciona uma representação mais refinada e contextualizada dos documentos, sendo amplamente empregado em tarefas de classificação de textos, incluindo a detecção de notícias falsas, onde a relevância e raridade das palavras são cruciais para uma análise precisa.

2.3 Algoritmos de Classificação

Os algoritmos de classificação desempenham um papel fundamental em processos de aprendizado de máquina, sendo projetados para automatizar a tarefa de categorizar dados em diferentes classes ou categorias. Esses algoritmos são aplicados em uma variedade de domínios, desde reconhecimento de padrões até diagnósticos médicos, incluindo a detecção de notícias falsas, no contexto deste projeto. Eles operam treinando em conjuntos de dados rotulados, onde as relações entre as entradas e suas classes correspondentes são conhecidas, permitindo que o algoritmo aprenda padrões e características associadas a cada classe.

A pesquisa de referência [4] adotou uma abordagem abrangente, aplicando 20 algoritmos distintos, tais como SVM, Regressão Logística, KNN, Naive Bayes, MLP, Random Forest, AdaBoost, XGBoost, CNN, BiLSTM, ELMO, BERT, DistilBERT, ALBERT, RoBERTa, BART, ELECTRA, XLNet, LLAMA e Falcon. Essa diversidade de algoritmos visou explorar amplamente as capacidades de diferentes técnicas de classificação para a detecção de notícias falsas. Na reprodução do estudo, foram aplicados seis algoritmos, incluindo SVM, Regressão Logística, KNN, Naive Bayes, MLP e Random Forest [10, 11], em uma abordagem mais focalizada, mantendo a robustez da análise e otimizando os recursos computacionais.

As Máquinas de Vetores de Suporte (SVM) são uma poderosa classe de algoritmos de aprendizado de máquina que se destacam pela capacidade de realizar tarefas de classificação e regressão. O SVM opera construindo um hiperplano ótimo que separa distintamente diferentes classes no espaço dimensional dos dados. Este hiperplano é escolhido de forma a maximizar a margem entre as classes, resultando em

uma abordagem robusta para lidar com conjuntos de dados complexos e não linearmente separáveis. A versatilidade do SVM permite não apenas a classificação eficaz, mas também a manipulação eficiente de conjuntos de dados de alta dimensionalidade, tornando-o uma escolha frequente em aplicações como a detecção de notícias falsas, onde a precisão e a capacidade de generalização são cruciais.

A Regressão Logística é um método amplamente empregado em análise estatística e aprendizado de máquina para realizar tarefas de classificação binária. Ao contrário do nome, a Regressão Logística não é destinada à análise de relações lineares contínuas, mas sim à modelagem da probabilidade de uma instância pertencer a uma determinada classe. Este algoritmo utiliza a função logística para mapear a soma ponderada das características de entrada a uma probabilidade no intervalo de 0 a 1. Com base nessa probabilidade, é possível realizar uma decisão de classificação, tornando a Regressão Logística particularmente valiosa em contextos como a detecção de notícias falsas, onde a interpretação probabilística das predições é essencial para uma análise crítica e precisa.

O algoritmo de K-Vizinhos Mais Próximos (KNN) é uma abordagem intuitiva e eficaz no campo da aprendizagem de máquina para tarefas de classificação. Sua lógica fundamenta-se na proximidade espacial entre pontos de dados, classificando uma instância desconhecida com base na maioria das classes dos seus k vizinhos mais próximos. Esse método não paramétrico destaca-se pela simplicidade e flexibilidade, sendo aplicável em uma variedade de contextos, inclusive na detecção de notícias falsas.

O algoritmo Naive Bayes é uma técnica de aprendizado de máquina baseada no teorema de Bayes, destacando-se pela sua simplicidade e eficácia em tarefas de classificação. A abordagem "naive" parte da suposição de independência condicional entre as características, o que simplifica significativamente os cálculos. Essa simplicidade, no entanto, não compromete a precisão, tornando o Naive Bayes uma escolha popular em aplicações como a detecção de notícias falsas. Ele utiliza a probabilidade condicional para estimar a probabilidade de uma instância pertencer a uma determinada classe, baseando-se nas frequências observadas no conjunto de treinamento.

As Redes Neurais de Perceptrons Multicamadas (MLP) representam uma classe poderosa de algoritmos de aprendizado de máquina, comumente utilizada em tarefas complexas, incluindo a classificação de dados. Essa arquitetura é composta por camadas de neurônios interconectados, cada qual realizando operações matemáticas em seus inputs ponderados e produzindo saídas que servem como inputs para as camadas subsequentes. A capacidade das MLPs de aprender representações não lineares complexas a partir dos dados as torna ideais para a detecção de padrões em conjuntos de dados diversos, sendo empregadas em aplicações como a identificação de notícias falsas. A utilização de técnicas de treinamento como o retropropagação permite ajustar os pesos da rede neural com base nos erros, contribuindo para a sua capacidade de generalização e adaptabilidade a problemas variados.

A Random Forest, ou Floresta Aleatória, é um algoritmo que combina múltiplas árvores de decisão durante o treinamento e toma decisões por meio da agregação de votos das árvores individuais. A aleatoriedade introduzida no processo, seja na seleção

de características ou na construção das árvores, contribui para a diversidade do modelo e reduz o risco de sobreajuste. Esse método é especialmente robusto em lidar com conjuntos de dados complexos e pode capturar padrões sutis, sendo aplicado em diversas áreas, incluindo a detecção de notícias falsas. A Random Forest destaca-se pela sua flexibilidade, precisão e capacidade de lidar com grandes volumes de dados.

2.4 Conjuntos de Dados

A pesquisa de referência [4] utilizou quatro conjuntos de dados (Liar, Covid, Isot, George McIntire). Para esta reprodução, optou-se por focar em três conjuntos específicos. Essa escolha foi orientada pelo tamanho do conjunto de dados e pela necessidade de manter a viabilidade do projeto, garantindo uma abordagem representativa.

O conjunto de dados Liar [6] é uma fonte fundamental para a pesquisa sobre detecção de Fake News, compreendendo cerca de 12,8 mil declarações rotuladas manualmente coletadas do PolitiFact.com. Essas declarações abrangem uma ampla gama de contextos, incluindo notícias, entrevistas televisivas e de rádio, e foram recolhidas ao longo do período de 2007 a 2016. Cada entrada no conjunto de dados não apenas inclui a declaração de notícias, mas também oferece informações detalhadas sobre o orador, contexto, afiliação partidária e uma classificação em uma das seis categorias, variando de "Pants-fire" a "True".

O conjunto de dados Covid [7] emerge como uma valiosa ferramenta de pesquisa ao se concentrar na detecção de Fake News relacionadas à pandemia da COVID-19. Composto por 10.700 entradas em inglês, esse conjunto foi coletado do Twitter e verificado por fontes confiáveis, como politifact.com e snopes.com. Cada entrada no conjunto é categorizada como notícia real ou falsa, apresentando uma compilação diversificada de informações que se mostram cruciais para a compreensão das narrativas e desinformação em torno da pandemia.

O conjunto de dados George McIntire (GM) [8] representa uma valiosa fonte de informações para estudos sobre notícias falsas. Composto por 11.000 artigos de notícias reais e 3.151 artigos de notícias falsas, provenientes de diversas fontes da mídia convencional, incluindo o New York Times, Wall Street Journal, Bloomberg e o Guardian, este conjunto abrange uma ampla gama de tópicos, como política, negócios, tecnologia e entretenimento. Cada artigo foi minuciosamente verificado por jornalistas, garantindo a confiabilidade dos rótulos no conjunto de dados.

2.4.1 Pré-processamento dos Dados

O pré-processamento de dados desempenha um papel crucial na preparação dos conjuntos de dados para análises subsequentes, influenciando diretamente a eficácia dos modelos de detecção de Fake News. Esse processo abrangente consiste em várias etapas, começando pela normalização, onde textos são ajustados para garantir uniformidade, como a conversão de letras maiúsculas para minúsculas. Em seguida, a tokenização divide o texto em unidades semânticas, geralmente palavras, facilitando a

análise individual. A remoção de stopwords é uma etapa vital para eliminar palavras comuns que não contribuem significativamente para o sentido da frase, reduzindo a dimensionalidade dos dados. A remoção de pontuação é essencial para focar nas palavras-chave e evitar distorções. Finalmente, o stemming simplifica as palavras ao reduzi-las às suas formas radicais, unificando variações vocabulares. Esse conjunto de técnicas de pré-processamento não apenas aprimora a qualidade dos dados, mas também otimiza a eficiência dos modelos de detecção, permitindo uma análise mais precisa e eficaz das características textuais relevantes.

3. Experimentos e Resultados

Esta seção detalha os experimentos conduzidos ao longo do desenvolvimento do projeto, desde o pré-processamento dos dados até a fase de classificação. A metodologia adotada foi dividida em etapas distintas, cada uma contribuindo para a análise abrangente dos métodos de detecção de notícias falsas.

3.1 Pré-processamento

O pré-processamento teve início com a carga dos conjuntos de dados utilizando a biblioteca Pandas. Para focar nas informações relevantes, foram descartadas todas as colunas que não contribuem diretamente para o entendimento das notícias, mantendo apenas as colunas de texto e rótulos (labels). Para visualização inicial, foi gerada uma imagem de nuvem de palavras, fornecendo insights visuais sobre as palavras mais frequentes.

Em seguida, uma série de técnicas de pré-processamento foi aplicada. A normalização converteu os textos para minúsculas, garantindo uniformidade. A remoção de pontuação, a tokenização, a eliminação de stopwords e o stemming visaram simplificar e padronizar o texto, preparando-o para análise mais eficiente.

3.2 Extração de Características

A fase de extração de características começou com a divisão do dataframe em conjuntos de treino e teste através da função ‘train_test_split’ da biblioteca scikit-learn, sendo 25% para teste e 75% treino, os 3 conjuntos de dados foram desbalanceados. A transformação dos dados em vetores numéricos foi realizada por meio da biblioteca 'sklearn.feature_extraction', resultando em matrizes esparsas para representar as características dos textos.

3.3 Classificação

A etapa de classificação incorporou a técnica de busca em grade (Grid Search), um processo sistemático para encontrar os melhores hiperparâmetros para um modelo de aprendizado de máquina. Foram utilizados seis algoritmos distintos: SVM, Random Forest, Naive Bayes, MLP, Regressão Logística e KNN. Os hiperparâmetros de cada

algoritmo foram definidos com base em uma busca extensiva, visando otimizar o desempenho de cada modelo.

3.3.1 Hiperparâmetros

- SVM (Support Vector Machine):
  - C: Controla a penalidade de erro. Valores menores indicam uma penalidade menor para classificações incorretas, levando a uma fronteira de decisão mais suave.
  - Gamma: Define o quão longe a influência de um único exemplo de treinamento alcança. Valores maiores levam a um alcance menor.
  - Kernel: Especifica o tipo de função kernel a ser usado na operação de transformação de dados em um espaço dimensional superior.

- Random Forest:
  - Bootstrap: Indica se a amostragem com reposição será usada ao construir árvores. Se definido como False, toda a amostra é usada para construir cada árvore.
  - Max_depth: Limita a profundidade máxima de cada árvore na floresta. Controla a complexidade do modelo.
  - Max_features: Define o número máximo de características a serem consideradas ao fazer uma divisão.
  - Min_samples_leaf: Define o número mínimo de amostras necessário para ser uma folha na árvore.
  - Min_samples_split: Define o número mínimo de amostras necessárias para dividir um nó.
  - N_estimators: Define o número de árvores na floresta.
  - Criterion: Define a medida usada para avaliar a qualidade de uma divisão.

- Naive Bayes:
  - Alpha: Controla o quanto de suavização será aplicado aos dados. Valores menores especificam mais suavização.
  - Fit_prior: Indica se as probabilidades a priori devem ser ajustadas com base nos dados.

- MLP (Multilayer Perceptron):
  - Activation: Define a função de ativação para a camada oculta. 'Relu' é uma função linear rectificada, e 'Logistic' é a função logística.
  - Solver: Define o algoritmo usado para otimizar os pesos da rede neural.

- Regressão Logística:
  - Solver: Define o algoritmo usado para resolver o problema de otimização. 'Liblinear' é um solver eficaz para conjuntos de dados pequenos.

    - Penalty: Define o tipo de regularização aplicada aos dados. 'None' significa nenhuma regularização.
    - C: Controla a força da regularização. Valores menores especificam uma regularização mais forte.

- KNN (K-Nearest Neighbors):
    - n_neighbors: Define o número de vizinhos considerados durante a classificação de um ponto. Um valor maior leva a uma superfície de decisão mais suave.

Método	Hiperparâmetros
SVM	C: [0.1, 1, 10, 100, 1000] Gamma: [1, 0.1, 0.01, 0.001, 0.0001] kernel: ['rbf']
Random Forest	Bootstrap: [True, False] max_depth: [5, 10, 30] max_features: [‘auto’, ‘log2’] min_samples_leaf: [1, 2, 4] min_samples_split: [2, 5, 10] n_estimators: [200, 400, 800] Criterion: [‘gini’, ‘entropy’]
Naive Bayes	Alpha: [0.1, 0.5, 1] fit_prior: [True, False]
MLP	Activation: [‘relu’, ‘logistic’] Solver: [‘adam’, 'lbfgs']
Regressão Logística	Solver : ['liblinear'] Penalty : ['none', 'l1', 'l2', 'elasticnet'] C : [100, 10, 1.0, 0.1, 0.01]
KNN	n_neighbors: [1-20]

Tabela 1. Hiperparâmetros considerados no aprendizado de máquina e modelos de conjunto

A busca em grade foi conduzida utilizando a classe ‘GridSearchCV’ da biblioteca scikit-learn e foi considerado os parâmetros scoring=’f1_micro’ (métrica usada para avaliar o desempenho do modelo durante a busca de hiperparâmetros), cv= 5 (especifica o número de dobras (folds) em que os dados serão divididos para treinamento e validação), n_jobs = -1 (controla o número de trabalhos em paralelo a serem executados durante a pesquisa de hiperparâmetros. Um valor de -1 indica que

todos os núcleos do processador serão utilizados, o que pode acelerar significativamente o processo de busca), verbose = 2 (controla o nível de detalhes das mensagens de saída durante o processo de busca de hiperparâmetros, um valor de 0 indica que nenhuma mensagem será exibida, um valor maior que 0 indica o número de mensagens detalhadas a serem exibidas), refit = True (especifica se o estimador deve ser reajustado usando os melhores hiperparâmetros encontrados após a conclusão da pesquisa).

3.4 Avaliação e Métricas

Para avaliar o desempenho dos modelos, foram consideradas as métricas de acurácia e a f1-score, proporcionando uma visão abrangente de sua eficácia na detecção de notícias falsas.

A acurácia é uma métrica usada para avaliar o desempenho de um modelo de classificação. Ela mede a habilidade do modelo em classificar corretamente os exemplos em todas as classes.

O F1-Score é uma métrica que combina precisão e recall em uma única pontuação. A precisão é uma métrica que mede a proporção de exemplos positivos que foram corretamente classificados como positivos em relação ao total de exemplos classificados como positivos pelo modelo, enquanto o recall mede a proporção de exemplos positivos que foram corretamente identificados pelo modelo em relação ao total de exemplos positivos presentes na base de dados. O F1-Score calcula a média harmônica dessas duas métricas, dando mais peso à menor delas. Isso torna o F1-Score uma métrica útil para avaliar o desempenho de um modelo de classificação, especialmente em situações em que há um desequilíbrio entre as classes.

A seguir, são apresentados os resultados detalhados para cada algoritmo e conjunto de dados específico:

	SVM	RF	NB	MLP	LR	KNN
TF	0.91	0.83	0.83	0.92	0.92	0.79
TF-IDF	0.93	0.83	0.83	0.92	0.93	0.71

Tabela 2. Acurácia obtida, no experimento, do conjunto de dados GM

	SVM	RF	NB	MLP	LR	KNN
TF	0.25	0.23	0.24	0.23	0.24	0.24
TF-IDF	0.25	0.24	0.24	0.22	0.24	0.24

Tabela 3. Acurácia obtida, no experimento, do conjunto de dados Liar

	SVM	RF	NB	MLP	LR	KNN
TF	0.92	0.89	0.92	0.91	0.92	0.78
TF-IDF	0.93	0.89	0.92	0.92	0.93	0.90

Tabela 4. Acurácia obtida, no experimento, do conjunto de dados COVID

	TF		TF-IDF	
	experimento	referência	experimento	referência
SVM	0.91	0.905	0.94	0.939
RF	0.84	0.901	0.84	0.906
NB	0.84	0.847	0.84	0.847
MLP	0.92	0.921	0.92	0.937
LR	0.92	0.919	0.94	0.936
KNN	0.79	0.796	0.77	0.594

Tabela 5. Comparação de F1-score entre o artigo de referência [4] e o resultado obtido do experimento com o conjunto de dados GM

	TF		TF-IDF	
	experimento	referência	experimento	referência
SVM	0.25	0.256	0.25	0.246
RF	0.23	0.246	0.24	0.236
NB	0.24	0.242	0.24	0.239
MLP	0.23	0.226	0.22	0.222
LR	0.24	0.245	0.24	0.242
KNN	0.24	0.219	0.24	0.228

Tabela 6. Comparação de F1-score entre o artigo de referência [4] e o resultado obtido do experimento com o conjunto de dados Liar

	TF		TF-IDF	
	experimento	referência	experimento	referência
SVM	0.92	0.920	0.93	0.928
RF	0.89	0.906	0.88	0.903
NB	0.92	0.916	0.92	0.916
MLP	0.90	0.908	0.91	0.919
LR	0.92	0.920	0.92	0.924
KNN	0.80	0.750	0.89	0.889

Tabela 7. Comparação de F1-score entre o artigo de referência [4] e o resultado obtido do experimento com o conjunto de dados COVID

Essas métricas oferecem uma compreensão abrangente do desempenho de cada algoritmo em diferentes conjuntos de dados, permitindo uma avaliação crítica e a seleção do modelo mais adequado para a detecção de notícias falsas.

4. Discussão

A análise das métricas apresentadas nas Tabelas 2, 3 e 4, proporciona uma visão aprofundada do desempenho dos algoritmos de detecção de notícias falsas nos conjuntos de dados avaliados. Uma observação inicial revela uma notável disparidade na acurácia dos algoritmos ao lidar com o conjunto de dados Liar em comparação com os conjuntos Covid e George McIntire. Essa discrepância sugere que os algoritmos testados foram originalmente concebidos para rótulos binários, uma vez que o conjunto Liar contém seis categorias distintas.

Analisando as acurácias obtidas nos conjuntos de dados utilizados, destaca-se que a combinação que obteve os resultados mais promissores foi a implementação do algoritmo de classificação Support Vector Machine (SVM) em conjunto com o extrator de características Term Frequency-Inverse Document Frequency (TF-IDF). O algoritmo Naive Bayes mostrou uma notável estabilidade, demonstrado pelas métricas F1-Score nas Tabelas 5, 6 e 7, nas implementações utilizando Term Frequency (TF) e Term Frequency-Inverse Document Frequency (TF-IDF), revelando uma consistência na sua capacidade de classificação. Essa estabilidade sugere que o Naive Bayes mantém sua eficácia ao longo das diferentes representações de características, indicando robustez na identificação de padrões e características relevantes para a detecção de notícias falsas.

As Tabelas 5, 6 e 7 destacam os resultados obtidos, na métrica F1-Score, tanto no experimento conduzido neste estudo quanto nos resultados apresentados no artigo de referência [4]. Ao analisar esses dados, observa-se uma notável semelhança nos desempenhos dos algoritmos em cada conjunto de dados avaliado. Esse alinhamento

entre os resultados confirma a validade e a consistência das metodologias adotadas neste estudo em relação às abordagens descritas na literatura. No entanto, ao examinar mais de perto, é possível identificar uma discrepância específica no desempenho do algoritmo Random Forest. Este algoritmo demonstrou uma performance ligeiramente inferior em comparação com os resultados reportados no artigo de referência [4]. Essa diferença merece uma investigação mais detalhada para compreender as possíveis causas subjacentes e identificar áreas de aprimoramento potencial no processo de implementação ou configuração do algoritmo.

5. Conclusão e Trabalhos futuros

Em resumo, a análise profunda do desempenho dos algoritmos de detecção de notícias falsas revela insights significativos sobre a complexidade dessa tarefa, especialmente ao considerar conjuntos de dados com múltiplas categorias, como o Liar. A disparidade na acurácia entre o Liar e outros conjuntos destaca a necessidade de adaptação dos algoritmos para lidar eficazmente com diferentes números de classes, sublinhando a complexidade inerente à diversidade de rótulos nas declarações de notícias.

Entre as diferentes combinações de algoritmos e extratores de características, a implementação do Support Vector Machine (SVM) em conjunto com o extrator de características Term Frequency-Inverse Document Frequency (TF-IDF) demonstrou consistentemente resultados promissores. Esse destaque sugere a eficácia dessa abordagem específica na detecção de notícias falsas e aponta para a importância da escolha criteriosa de algoritmos e técnicas de extração de características, adaptadas às nuances específicas de cada conjunto de dados.

A análise das métricas apresentadas, proporciona uma visão aprofundada do desempenho dos algoritmos de detecção de notícias falsas nos conjuntos de dados avaliados. Uma observação inicial revela uma notável disparidade na acurácia dos algoritmos ao lidar com o conjunto de dados Liar em comparação com os conjuntos Covid e George McIntire.

Considerando a dinâmica em constante evolução da disseminação de notícias falsas, identificamos várias oportunidades para pesquisas futuras aprimorarem e expandirem este estudo. Essas direções incluem aprofundar a análise de conjuntos de dados específicos, incorporar técnicas avançadas de Processamento de Linguagem Natural (PLN), explorar abordagens ensemble, integrar fontes de mídia social, considerar informações temporais, estudar viés e neutralidade, e desenvolver ferramentas práticas para a identificação rápida e eficaz de notícias falsas. Essas sugestões destacam a necessidade contínua de inovação e adaptação diante do desafio dinâmico e em constante evolução da detecção de notícias falsas.

Além disso, buscando promover a transparência e replicabilidade dos nossos experimentos, disponibilizamos o código-fonte completo do projeto no repositório https://github.com/JosenildoVicente/estudo-comparativo-deteccao-fake-news.

Referências

[1] Mais de 60% da população mundial está ativa nas redes sociais, aponta relatório. **Diário do Nordeste**, 2023. Disponível em: https://diariodonordeste.verdesmares.com.br/ultima-hora/tecnologia/mais-de-60-da-populacao-mundial-esta-ativa-nas-redes-sociais-aponta-relatorio-1.3395128. Acesso em: 30, jan. 2024.

[2] FERREIRA, Ednael. O impacto das redes sociais no consumo de notícias. **Gorila Blog**, 2023. Disponível em: https://gorila.com.br/blog/o-impacto-das-redes-sociais-no-consumo-de-noticias. Acesso em: 30, jan. 2024.

[3] Fake News disseminam mais rápido que notícias verdadeiras. **Instituto Qualibest**, 2021. Disponível em: https://www.institutoqualibest.com/marketing/fake-news-disseminam-mais-rapido-que-noticias-verdadeiras. Acesso em: 30, jan. 2024.

[4] FARHANGIAN, Faramarz; CRUZ, Rafael M.O.; CAVALCANTI, George D.C. Fake news detection: Taxonomy and comparative study. **ScienceDirect**, 2023. Disponível em: https://doi.org/10.1016/j.inffus.2023.102140. Acesso em: 28 jan. 2024.

[5] GIORDÃO, Mariana; OLIVEIRA, Mila de. O impacto das fake news na sociedade. **G&A Comunicação Blog**, 2020. Disponível em: https://www.geacomunicacao.com.br/insights/o-impacto-das-fake-news-na-sociedade/. Acesso em: 02, fev. 2024.

[6] WANG, William Y. "Liar, Liar Pants on Fire": A New Benchmark Dataset for Fake News Detection. **ArXiv preprint**, 2017. Disponível em: arXiv:1705.00648. Acesso em: 02, fev. 2024.

[7] PARTH, Parth; SHARMA, Shivan; PYKL, Srinivas; GUPTHA, Vineeth; KUMARI, Gitanjali;AKHTAR, M.S.; EKBAL, Asif; DAS, Amitava; CHAKRABORTY, Tanmoy. Fighting an infodemic: COVID-19 fake news dataset. **ArXiv preprint**, 2020. Disponível em: arXiv:2011.03327. Acesso em: 02, fev. 2024.

[8] MCINTIRE, George. Mcintire fake news dataset. **Github**, 2017. Disponível em: https://github.com/lutzhamel/fake-news. Acesso em: 02, fev. 2024.

[9] FONSECA, Camilla. Introdução a Bag of Words e TF-IDF. **Medium**, 2020. Disponível em: https://medium.com/turing-talks/introdu%C3%A7%C3%A3o-a-bag-of-words-e-tf-idf-43a128151ce9. Acesso em: 02, fev. 2024.

[10] PELLANDA, Bruno. Modelos de Machine Learning: uma comparação entre os modelos — Parte 1. **Medium**, 2021. Disponível em: https://medium.com/gbtech/modelos-de-machine-learning-uma-compara%C3%A7%C3%A3o-entre-os-modelos-parte-1-c772661c7163. Acesso em: 03, fev. 2024.

[11] PELLANDA, Bruno. Modelos de Machine Learning: Uma comparação entre os modelos — Parte 2. **Medium**, 2021. Disponível em:

https://medium.com/gbtech/modelos-de-machine-learning-uma-compara%C3%A7%C3%A3o-entre-os-modelos-parte-2-3b79cd2c84ab. Acesso em: 03, fev. 2024.

[12] ANTONIO, Livia G. Redes Sociais e Fake News: como a combinação impacta a sociedade?. **Politize!**, 2023. Disponível em: https://www.politize.com.br/redes-sociais-e-fake-news/. Acesso em: 31, jan. 2024.

[13] BATISTA, Rafael. Fake News. **Mundo Educação**. Disponível em: https://mundoeducacao.uol.com.br/curiosidades/fake-news.htm. Acesso em: 31, jan. 2024.

[14] OZBAY, Feyza Altunbey; ALATAŞ, Bilal. Fake news detection within online social media using supervised artificial intelligence algorithms. **ScienceDirect**, 2020. Disponível em: https://doi.org/10.1016/j.physa.2019.123174. Acesso em: 1, fev. 2024.

[15] ABDULRAHMAN, Awf ; BAYKARA, Muhammet. Fake News Detection Using Machine Learning and Deep Learning Algorithms. **IEEE Xplore**, 2021. Disponível em: https://doi.org/10.1109/ICOASE51841.2020.9436605. Acesso em: 1, fev. 2024.