Classificação de *Fake News* utilizando o *dataset* LIAR

Guilherme Veiga Santos Pinto
guilhermeveigarj@edu.univali.br
Escola Politécnica, Universidade do Vale do Itajaí
Itajaí, SC, Brazil

Ramices dos Santos Silva
ramices@edu.univali.br
Escola Politécnica, Universidade do Vale do Itajaí
Itajaí, SC, Brazil

Rudimar L. Scaranto Dazzi
rudimar@univali.br
Escola Politécnica, Universidade do Vale do Itajaí
Itajaí, SC, Brazil

Raimundo C. Ghizoni Teive
rteive@univali.br
Escola Politécnica, Universidade do Vale do Itajaí
Itajaí, SC, Brazil

Anita M. da Rocha Fernandes
anita.fernandes@univali.br
Escola Politécnica, Universidade do Vale do Itajaí
Itajaí, SC, Brazil

Wemerson Delcio Parreira
wemerson.delcio@puc-campinas.edu.br
Escola Politécnica, Pontifícia Universidade Católica de Campinas
Campinas, SP, Brazil

ABSTRACT

Society grapples with an overwhelming and ceaseless flow of information, posing challenges to a media environment already afflicted by eroding trust in news. The use of Supervised Learning models for Fake News classification is widespread, yet their effectiveness hinges on the quality of labeled data. Constructing datasets that encompass the intricate nuances of disinformation across diverse contexts remains a formidable task. This study presents a comparative analysis of various Supervised Learning models for detecting and classifying misinformation. Leveraging the LIAR dataset, which employs six different classes to characterize the veracity of statements, our findings align with the accuracy benchmarks established by the LIAR authors. Specifically, the logistic regression model with stemming achieves an accuracy of 25%. The study suggests potential enhancements through the application of Deep Learning techniques, revealing a positive correlation between accuracy and the number of training epochs. Despite current accuracy levels, notably lower than datasets with binary classifiers, it is crucial to underscore the meticulous manual verification and annotation process executed by the LIAR dataset authors.

KEYWORDS

Fake News, Processamento de Linguagem Natural, Machine Learning, Deep Learning

1 INTRODUÇÃO

A sociedade está sobrecarregada com um fluxo elevado e ininterrupto de informações, o que gera dificuldades em tarefas de discernimento entre material confiável e conteúdo que procura enganar através da desinformação — seja intencionalmente ou não —[1]. Isso gera implicações problemáticas para um ambiente de mídia onde a confiança nas notícias já está reduzida. Segundo o Relatório de Notícias Digitais de 2022, elaborado pelo Instituto Reuters, a confiança nas notícias está diminuindo, com apenas 42% das pessoas da amostra global dizendo que confiam na maioria das notícias na maior parte do tempo. Em relação aos brasileiros, esta confiança é de 48%, sendo menor que a metade [2].

Modelos de Aprendizado de Máquina como *Large Language Models* (LLMs) são especializados em prever a próxima palavra em uma sequência que pode ser usada para geração de texto e tradução [3]. Um LLM particularmente notável é o *Generative Pre-trained Transformer* (GPT). Suas versões mais recentes GPT-3 e GPT-4 podem gerar linguagem natural e executar uma ampla gama de tarefas de processamento natural de linguagem, como geração de texto, tradução automática e resposta a perguntas [4].

Mais recentemente, uma variante do GPT-3 chamada ChatGPT foi treinada com sucesso por meio da interação humana para se envolver em conversas realistas [5]. Esta variante se tornou cada vez mais popular desde seu início em novembro de 2022, registrando 100 milhões de usuários mensais em janeiro de 2023 [6], e consequentemente, gerando desinformação.

Machine Learning (ML) é um campo da Inteligência Artificial que engloba uma variedade de métodos, técnicas e ferramentas para construir sistemas inteligentes, se enquadrando no paradigma de reconhecimento de padrões, para identificar características repetidas em uma amostra de dados, usando processos estatísticos e computacionais. Esses padrões atendem a duas funções principais: (i) fazer previsões sobre eventos futuros (análise preditiva) e (ii) descobrir percepções dos dados (análise descritiva). Dependendo do modo de aprendizado e do processo de obtenção de padrões, existem três famílias principais de técnicas de ML: Aprendizado Supervisionado, Não Supervisionado e por Reforço [7].

O Aprendizado Supervisionado busca desenvolver modelos a partir de dados de treinamento rotulados que permitem prever os rótulos de dados não vistos ou futuros, enquanto a Aprendizagem Não Supervisionada refere-se a técnicas que lidam com dados não rotulados ou não estruturados. A Aprendizagem Supervisionada é a abordagem mais amplamente empregada para a identificação automática de desinformação. Assim, a identificação da desinformação é geralmente modelada como um problema de classificação binária. No entanto, a desinformação flui em tons de cinza, não em preto e branco, tornando uma classificação binária as vezes insuficiente [6].

O desempenho do Aprendizado Supervisionado depende diretamente da qualidade dos dados rotulados, que geralmente representam situações, dificultando a extensão dos modelos para outros domínios semelhantes. Essa limitação é ainda mais perceptível quando aplicada à detecção automática de desinformação, pois é desafiador construir conjuntos de dados com qualidade suficiente para cobrir as nuances da desinformação em contextos heterogêneos [8].

Ao mesmo tempo, os estudos que trabalham diretamente na detecção automática de desinformação com Aprendizagem Não Supervisionada são escassos [9–11]. A maioria dos estudos utiliza o Aprendizado Não Supervisionado de forma complementar ao Aprendizado Supervisionado, utilizando uma abordagem Semi-Supervisionada [12–16].

Neste sentido, este trabalho apresenta um estudo comparativo entre modelos de Aprendizagem Supervisionada, para detecção e classificação de desinformação. Para isso são usados *datasets* que empregam o conceito de desinformação fluindo em tons de cinza. Este conceito pode ser empregado através da utilização de mais de dois classificadores para determinar o nível de veracidade de uma informação. Dentre os *datasets* que trabalham com classificação de informação de maneira não-binária, estão NELA-GT-2018 [17], CREDBANK [18], BUZZFACE [19], BUZZFEEDNEWS [20], FEVER [21] e LIAR [22], sendo o último utilizado neste trabalho.

As demais seções deste trabalho está organizado como se segue. Na Seção 2 é apresentado uma revisão dos algoritmos e *dataset* empregados. A Seção 3 apresenta uma visão detalhada da metologia empregada. Os resultados são discutidos na Seção 4. Finalmente, este trabalho é concluído com a Seção 5.

2 FUNDAMENTAÇÃO TEÓRICA

Nesta seção é apresentada uma revisão dos algoritmos usados para o desenvolvimento deste trabalho, adicionalmente, é apresentado uma descrição do *dataset* usado.

2.1 Processamento de Linguagem Natural

O Processamento de Linguagem Natural (PLN) utiliza técnicas de linguística computacional para analisar textos em um idioma concreto [23] através de três etapas: pré-processamento do texto, extração de características e representação numérica.

O pré-processamento do texto consiste em limpar o texto e eliminar elementos supérfluos, deixando somente as informações úteis. Este procedimenhto é possivel a partir da Tokenização, que divide o texto bruto em fragmentos, normalmente palavras; Eliminação de palavras comuns que são irrelevantes ao conjunto de dados (*Stopwords*); Stemização (*Stemming*), procedimento responsável por reduzir as palavras do texto à sua palavra raiz; e Lematização que transforma as palavras em sua forma base ou lema [7].

A extração de características busca identificar e selecionar os traços básicos dos dados textuais resultantes. Algumas das técnicas mais utilizadas para a extração de características são: Etiquetagem *Part-Of-Speech* (POS) para identificar categorias léxicas; Reconhecimento de Entidades Nomeadas (*Named-Entity Recognition*) para identificar entidades dentro do texto; E Saco de Palavras (*Bag of Words*), para representar unidades linguísticas em função de sua frequência de aparição [7]. Outra técnica de extração de características é a Análise de Sentimentos, com o objetivo é captar automaticamente os sentimentos, opiniões, emoções ou atitudes subjacentes em um texto [24].

Já a representação numérica envolve a criação de uma codificação numérica do texto para que outros algoritmos de *Machine Learning* possam realizar cálculos. Existem muitas técnicas para obter esta representação, sendo o *embedding* de palavras uma das mais utilizadas atualmente, permitindo capturar parcialmente a semântica do texto. Uma vez que um documento é representado numericamente, pode-se aplicar Técnicas de *Machine Learning* e *Deep Learning* para classificar ou prever texto [7].

2.2 Machine Learning e Deep Learning

Machine Learning é um campo da Inteligência Artificial que engloba uma variedade de métodos, técnicas e ferramentas para construir sistemas inteligentes, se enquadrando no paradigma de reconhecimento de padrões para identificar características repetidas em uma amostra de dados, usando processos estatísticos e computacionais. Esses padrões atendem a duas funções principais: fazer previsões sobre eventos futuros (análise preditiva) e descobrir percepções dos dados (análise descritiva). Dependendo do modo de aprendizado e do processo de obtenção de padrões, existem três famílias principais de técnicas de *Machine Learning*: Aprendizado Supervisionado, Não Supervisionado e por Reforço [7].

O *Deep Learning*, baseado em redes neurais, é geralmente classificado como Aprendizagem Supervisionada, mas também pode ser aplicado a problemas de Aprendizagem Não Supervisionada ou por Reforço. Desde a última década vem sendo uma tendência dominante [25], insere-se, em princípio, no âmbito da aprendizagem supervisionada, embora a sua aplicação tem sido utilizada em outros paradigmas. O seu modelo computacional é inspirado no córtex humano, e incorpora múltiplas camadas de processamento para capturar relacionamentos complexos em grandes conjuntos de dados. Dentro dele, observa-se diferentes tipos de algoritmos, como Redes Neurais Convolucionais, que são redes neurais especializadas em processamento de dados com um estrutura regular (como imagens); Redes Neurais Recorrentes, que permitem ciclos de *feedback* no seu cálculo e são aplicadas à dados sequenciais, como séries temporais e texto; e Transformadores, que aprendem a identificar seções relevantes de sequências aplicando modelos de atenção, e consequentemente sendo úteis com dados textuais [7].

2.3 Regressão Logística

Desenvolvida pelo estatístico David Cox em 1958, a Regressão Logística é um modelo de regressão onde a variável de resposta Y é categórica. A regressão logística permite estimar a probabilidade de uma resposta categórica com base em uma ou mais variáveis preditoras (X). Isto permite dizer que a presença de um preditor aumenta, ou diminui, a probabilidade de um determinado resultado em uma porcentagem específica [26].

Diferentemente da Regressão Linear, encontrar as melhores estimativas requer melhorar repetidamente as estimativas aproximadas até que a estabilidade seja alcançada. Isso é feito facilmente em um computador, e há muitos pacotes de software estatísticos que realizam regressão logística, mas torna a regressão logística menos compreensível e mais uma abordagem de "caixa preta" para muitos pesquisadores [27].

2.4 Naive Bayes

O Naive Bayes é uma técnica de classificação baseada no Teorema de Bayes, com uma suposição de independência entre os preditores dada por:

$$P(c|x) = \frac{P(x|c)P(c)}{P(x)} \quad (1)$$

em que $P(c|x)$ e $P(x|c)$ são as probabilidades a posteriori; $P(c)$ é a probabilidade a priori; $P(x)$ é a probabilidade a prior do preditor. Em termos simples, um classificador Naive Bayes assume que a presença de uma característica específica em uma classe não está relacionada à presença de qualquer outro recurso [28]. Este modelo é amplamente utilizado na indústria de classificação de texto, além de ser usado principalmente para *clustering*, tendo seu propósito de classificação dependente da probabilidade condicional acontecer.

2.5 Support Vector Machine

Outra técnica de Aprendizagem de Máquina amplamente utilizada é o *Support Vector Machine* (SVM), que são modelos de Aprendizagem Supervisionada associados com algoritmos de aprendizagem. Estes analisam dados utilizados em classificação e regressão. Além de realizar classificação linear, os SVMs podem realizar com eficiência uma classificação não linear, utilizando o truque do kernel para mapear implicitamente suas entradas em espaços de recursos de alta dimensão [29].

O objetivo de um SVM é pegar grupos de observações e construir limites para prever a qual grupo as observações futuras pertencem com base em suas medições. Os diferentes grupos que devem ser separados serão chamados de "classes". SVMs podem lidar com qualquer número de classes, bem como observações de qualquer dimensão, podendo assumir quase qualquer formato (incluindo linear, radial e polinomial, entre outros), e geralmente são flexíveis o suficiente para serem usados em praticamente qualquer empreendimento de classificação que o usuário decida realizar [30].

2.6 Passive-Agressive Classifier

Classificadores Passivo-Agressivos (*Passive-Agressive Classifiers*) trabalham de acordo com a atualização da sua classificação, quando por exemplo, ocorre uma classificação incorreta em um dado recém-visualizado, ou se uma margem predeterminada não for excedida por seu valor. Classificadores Passivo-Agressivos provaram ser um método muito eficaz e popular método de Aprendizagem Supervisionada para resolver muitos problemas do mundo real, sendo bastante utilizado nos casos em que há necessidade de verificar dados como notícias e mídias sociais. O princípio básico deste algoritmo é visualizar os dados, aprender com eles e descarta-los sem a necessidade de armazená-los. Para cada classificação incorreta, o algoritmo reage agressivamente atualizando os valores, enquanto para uma classificação correta, reage de forma preguiçosa ou passiva, condizente com o seu nome [31].

2.7 Árvore de Decisão

A Árvore de Decisão é um algoritmo que representa as escolhas e seus resultados em estrutura de árvore. Conforme observado na Figura 1, os nós no gráfico representam um evento ou escolha e as bordas do gráfico representam as regras de decisão ou condições. Cada árvore consiste em nós e ramos, onde cada nó representa atributos em um grupo a ser classificado e cada ramo representa um valor que o nó pode levar [29].

2.8 Random Forest

Florestas Aleatórias (*Random Forest*) são uma modificação do Algoritmo de Bagging, que constroem uma grande coleção de árvores

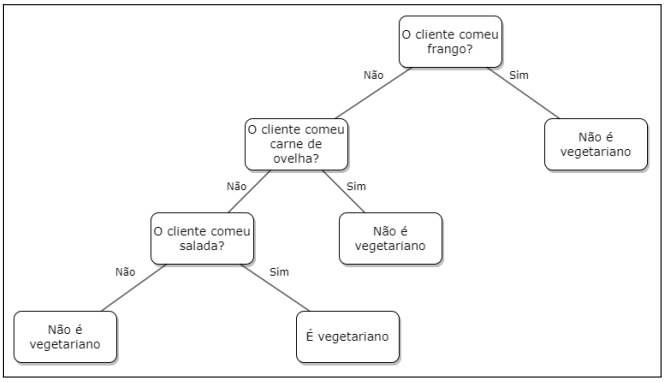

Figura 1: Visualização da Árvore de Decisão

descorrelacionadas e se tornaram um algoritmo de aprendizado "pronto para uso" muito popular, apresentando um bom desempenho preditivo. Estas são construídas com base nos mesmos princípios fundamentais das árvores de decisão e do ensacamento, que introduz um componente aleatório no processo de construção da árvore e reduz a variação da previsão de uma única árvore, melhorando o seu desempenho preditivo [32].

No entanto, as árvores no ensacamento não são completamente independentes umas das outras, uma vez que todos os preditores originais são considerados em cada divisão de cada árvore. Em vez disso, árvores de diferentes amostras de *bootstrap* normalmente têm estruturas semelhantes entre si (especialmente no topo da árvore) devido aos relacionamentos subjacentes. Essa característica é conhecida como correlação de árvore e evita que o ensacamento reduza de maneira ideal a variância dos valores preditivos. Para reduzir ainda mais a variância, é preciso minimizar a quantidade de correlação entre as árvores. Isto pode ser conseguido injetando mais aleatoriedade no processo de crescimento das árvores [32].

2.9 Bidirectional Encoder Representations for Transformers – BERT

BERT é um modelo de *Deep Learning*, sendo fundamentalmente uma pilha de transformadores de camadas codificadoras, que consistem de múltiplas "cabeças" de autoatenção. Para cada *token* de entrada em uma sequência, cada cabeça calcula a chave, o valor e os vetores de consulta utilizados para criar uma representação ponderada. As saídas de todos os cabeçotes na mesma camada são combinadas e executadas através de um sistema totalmente conectado em camadas, onde cada camada é envolvida com uma conexão de salto, e é seguida pela sua normalização. O fluxo de trabalho convencional do BERT consiste de duas etapas: Pré-Treinamento e Ajuste Fino (*Fine-Tunning*) [33].

O Pré-Treinamento usa duas tarefas auto-supervisionadas: Modelagem de Linguagem Mascara, que prevê aleatoriamente *tokens* de entrada mascarados, e a previsão da próxima frase, que prevse duas sentenças de entrada são adjacentes entre si. No Ajuste Fino para aplicações *downstream*, uma ou mais camadas totalmente conectadas são normalmente adicionadas no topo da camada final do codificador. BERT é o modelo baseado em transformador mais conhecido, e obteve resultados de última geração em vários *benchmarks* [33].

2.10 LIAR dataset

O *dataset* LIAR consiste em 12.836 declarações curtas e rotuladas manualmente, com duração de uma década, em vários contextos do projeto PolitiFact. O PolitiFact é um projeto sem fins lucrativos, com o intuito de realizar checagem de fatos em diversas declarações, fornecendo relatórios de análise detalhados e links para obter documentos de origem para cada caso. Esse o conjunto de dados também pode ser usado para pesquisas de verificação de fatos, possuindo uma ordem de magnitude maior do que os maiores conjuntos de dados públicos de notícias falsas de tipo semelhante [22].

O LIAR constitui de declarações curtas da política estadunidense, rotuladas por veracidade, assunto, contexto e local, orador, estado, partido e história anterior. Adicionalmente, em contraste à conjuntos de dados de *crowdsourcing*, as instâncias no LIAR são coletados de uma forma mais natural e fundamentada em contexto, como debate político, anúncios de TV, postagens no Facebook, tweets, entrevistas, comunicados de imprensa, etc. Para cada caso, o rotulador fornece uma análise detalhada para fundamentar cada julgamento, e os links para todos os documentos comprovativos também são fornecidos [22].

Diferente de *datasets* de *Fake News* com classificadores binários, o LIAR trabalha com seis rótulos refinados para as classificações de veracidade: *pants-fire, false, barelytrue, half-true, mostly-true, and true.* A distribuição de rótulos no conjunto de dados LIAR é relativamente bem equilibrada, conforme a figura 2. Exceto por 1.050 declarações do tipo *pants-fire*, as instâncias para todos os outros rótulos variam de 2.063 para 2.638 declarações [22]. Cada declaração possui uma *id* atrelada a ela, uma *label* correspondente ao seu grau de veracidade, o *statement* correspondente à declaração em si, o contexto e o *speaker* sendo o responsável pela declaração. O LIAR também registra algumas informações atreladas à aquele declarador, como o seu cargo político *(job title)*, estado de residência *(state info)*, partido político *(party affiliation)*, e um contador à respeito de quantas declarações em cada um dos seis rótulos de veracidade foram feitas pelo declarador.

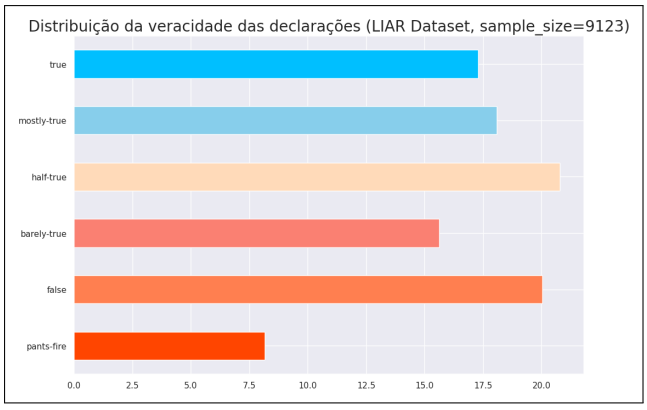

Figura 2: Gráfico de Distribuição de declarações do *dataset* LIAR

3 METODOLOGIA

Neste trabalho, foram analisados sete algoritmos de aprendizagem supervisionada, sendo seis deles de *Machine Learning* e um de *Deep Learning*, junto com os *datasets* LIAR e *Fake News Detection* [34], com o objetivo de avaliar a acurácia desses modelos. Anteriormente a etapa de treinamento, também foi feita uma etapa de pré-processamento dos dados contidos no *dataset* LIAR. A linguagem Python foi utilizada juntamente com a biblioteca PyTorch [35] e Scikit-learn [36], que fornecem diversas ferramentas para treinar e classificar modelos de Aprendizagem Profunda. O modelo implementado neste trabalho foi processado em um ambiente Google Colab [37], que permite escrever código em Python diretamente pelo navegador, com processamento na nuvem.

3.1 Seleção de Modelos

Para a seleção de modelos, foi feito levantamento na literatura procurando encontrar modelos de Aprendizagem Supervisionada mais utilizados na tarefa de detecção de desinformação. A partir deste levantamento, conforme observado em [7, 8, 10, 31], em modelos *Machine Learning*, observa-se que os modelos Regressão Logística, Naive Bayes, *Support Vector Machines*, *Passive-Agressive*, Árvores de Decisão e Random Forest, são os mais utilizados nesta tarefa. Quanto aos modelos de *Deep Learning*, o BERT (*Bidirectional Encoder Representations from Transformers*), é bastante utilizado em tarefas que lidam com processamento de linguagem natural.

3.2 Pré-processamento

A etapa de pré-processamento foi realizada com o intuito de preparar os dados do *dataset* LIAR, de tal forma que os treinamentos pudessem ocorrer nos modelos de Aprendizagem Supervisionada, em seus respectivos cenários de avaliação. Primeiramente foi feita uma remoção da extensão JSON que se encontrava atrelada em cada uma das instancias correspondentes ao ID do *Statement*. Também foi criada uma nova coluna na base chamada *Numeric Label*, sendo esta uma cópia da coluna Label, porém com um número atrelado a classe de veracidade do *Statement*. Esta coluna foi criada devido a uma questão interna do modelo BERT, que treina múltiplos classificadores de uma base em formato de número inteiro.

Em seguida, foram feitos os procedimentos padrões de processamento de linguagem natural, para todos os cenários de treinamento. Foi feita uma verificação e remoção de entradas vazias para não serem utilizadas nestes cenários, seguida pela remoção de palavras comuns (*Stopwords*) contidas na coluna *Statement*, visando reduzir o ruído dos dados analisados. Dando sequência, a etapa de Vetorização foi responsável por transformar os textos em sequências numéricas, para alimentar os modelos, e a etapa de Tokenzinação foi responsável por quebrar o texto em pedaços menores (*tokens*).

3.3 Treinamento e Avaliação dos Modelos

Os treinamentos foram realizados no Google Colab, com as configurações padrões de cada um dos modelos de Aprendizagem Supervisionada. Apenas os cenários utilizando o modelo BERT em que alterações foram implementadas, sendo elas configurações do treinamento para receber um multi-classificador de seis classes, em vez de duas, e a quantidade de *Epochs* que foi sendo incrementada. Em um outro cenário de Regressão Logística, foi utilizada uma abordagem diferente em que avalia-se a união das colunas *Speaker* com *Statement*, e também realiza-se o procedimento de stemização

(*Stemming*), procedimento responsável por reduzir as palavras do texto à sua palavra raiz.

Exceto pelos modelos BERT e o de Regressão Logística utilizando *Stemming*, e unindo as tabelas *Speaker* e *Statement*, todos os demais modelos foram treinados em um mesmo pipeline de código do Google Colab. Estes cenários também foram replicados para receber o *dataset* de classificação binária *Fake News Detection* [34], com o intuito de realizar uma análise comparativa da acurácia em todos esses modelos através de seus resultados. Para os modelos de *Machine Learning*, o tempo médio de treinamento levou entre segundos à minutos, apresentando um tempo elevado (quando comparado aos demais) somente na etapa de configuração do cenário e de pré-processamento destes dados. Já no modelo BERT, por ser um modelo de *Deep Learning*, seu tempo de treinamento aumentou conforme o incremento da quantidade de *epochs* utilizadas em cada um dos cenários, sendo seu tempo de treinamento entre 10 minutos à 3 horas com as configurações padrões de máquinas oferecidas pela versão gratuita do Google Colab.

4 RESULTADOS

No primeiro cenário, foi avaliado os modelos de *Machine Learning* e *Deep Learning* utilizando a acurácia como métrica, que consiste na quantidade de acertos do modelo divido pelo total da amostra. A Tabela 1 apresenta um comparativo com os resultados obtidos pelos autores do *dataset* LIAR [22], que também avaliaram o *dataset* em modelos de Aprendizagem Supervisionada. Este comparativo foi realizado com o intuito de avaliar tantos os mesmos modelos utilizados pelos autores do *dataset* LIAR, como também avaliar os modelos que são amplamente utilizados em implementações para detecção e classificação de *Fake News*.

Tabela 1: Tabela comparativa.

Modelo	Acurácia Este Trabalho	Acurácia (LIAR) [22]
Regressão Logística	0,24	0,24
Regressão Logística (*Stemming*, *Speaker + Statement*)	0,25	-
Naive Bayes	0,24	-
Support Vector Machine	0,24	0,25
Passive-Agressive	0,21	-
Árvore de Decisão	0,20	-
Random Forest	0,25	-
Majority	-	0,20
Bi-LSTMs	-	0,23
BERT (2 Epochs)	0,20	-
BERT (10 Epochs)	0,22	-
BERT (20 Epochs)	0,23	-
CNNs (Kim, 2014)	-	0,27

Pode-se observar na Tabela 1 uma média de acurácia bastante próxima com os testes realizados pelos outros autores, tendo como destaque no trabalho deles, o modelo de CNN contendo uma acurácia de 27% para classificação de *Fake News*. Neste trabalho, o modelo de Regressão Logística, quando utilizado a técnica de *Stemming* e treinado com as informações das colunas *Speaker* e *Statement*, demonstrou os melhores resultados apresentando uma acurácia de 25%. A Tabela 1 apresenta uma sumarização dos resultados, em que observar-se que, por mais que a quantidade de modelos de *Machine Learning* e *Deep Learning* tenha sido ampliada nesta análise comparativa, com o intuito de agregar a literatura, a média de acurácia entre esses modelos mostrou-se muito próxima dentre os 10 modelos avaliados nos dois trabalhos.

A partir dos resultados obtidos no primeiro cenário, foi conduzida uma segunda análise comparativa dos modelos de Aprendizagem Supervisionada, utilizados neste trabalho com o *dataset* LIAR, mas desta vez comparando-os quando treinados com o *dataset Fake News Detection* [34] que aborda uma classificação binária da desinformação. Nesta análise, houve uma grande discrepância nos valores da acurácia quando comparados com o *dataset* proposto, onde obteve-se uma média geral de 23% de acurácia nos modelos treinados com o *dataset* LIAR, enquanto esta média foi de 96% nos modelos treinados com o *dataset Fake News Detection*. A Tabela 2 apresenta os resultados comparativos entre estes dois *datasets* .

Como destaque, os modelos *Support Vector Machine*, *Passive-Aggressive*, Árvore de Decisão e *Random Forest*, apresentaram os melhores resultados na classificação binária da desinformação, evidenciando uma acurácia de 99%, contra os 25% obtidos na classificação não-binária com os modelos *Random Forest* e de Regressão Logística, utilizando *Stemming* com as colunas *Speaker* e *Statement*. Entretanto, é importante reforçar o fato de que o *dataset* LIAR trabalha com aproximadamente 12 mil declarações, divididas em seis classes, enquanto o *dataset Fake News Detection* trabalha com 44 mil declarações, dividas em apenas duas classes.

Outro fator relevante,é o fato do *dataset* LIAR trabalhar com dados que foram manualmente verificados e anotados pelo autor, em conjunto com a informação de onde estes dados foram coletados, diferente do *dataset Fake News Detection*, em que tanto a verificação dos dados, quanto o de local onde foi feita a coleta deste dados, não é informada pelo autor da base.

Tabela 2: Tabela comparativa entre os resultados do *dataset* LIAR com os resultados do *dataset Fake News Detection*.

Modelo	Acurácia Este Trabalho	Acurácia (*Fake News*) (*Detection*)
Regressão Logística	0,24	0,98
Regressão Logística (Stemming, Speaker + Statement)	0,25	0,98
Naive Bayes	0,24	0,97
Support Vector Machine	0,24	0,99
Passive-Aggressive	0,21	0,99
Árvore de Decisão	0,20	0,99
Random Forest	0,25	0,99
BERT (2 Epochs)	0,20	0,82
BERT (10 Epochs)	0,22	-
BERT (20 Epochs)	0,23	-

5 CONCLUSÃO

Neste trabalho, foram analisados sete algoritmos de aprendizagem supervisionada, sendo seis deles de *Machine Learning* e um de *Deep Learning*, procurando avaliar o seu desempenho para a classificação de desinformação com o *dataset* LIAR, comparar os resultados deste trabalho com os resultados obtidos pelos autores originais do dataset, e por final, compara-lo com o *dataset Fake News Detection*, utilizando os mesmos modelos. Os modelos *Random Forest* e de Regressão Logística, quando aplicado com *stemming* e treinado com as colunas *Speaker* e *Statement* do *dataset* LIAR, apresentaram os melhores resultados neste estudo comparativo com uma acurácia de 25%, apesar do modelo BERT apresentar uma margem maior de aumento de precisão conforme o número de *epochs* aumenta nos treinamentos.

Por mais que outros modelos de Aprendizagem Supervisionada foram apresentados e avaliados neste estudo comparativo, a média de precisão entre eles mostrou-se próxima aos modelos utilizados pelos autores do LIAR. Entretanto, ainda é uma precisão considerada baixa quando comparada à um *dataset* de *Fake News* que trabalha com classificação binária, e que possui muitos mais dados em sua base, apresentando uma discrepância de 73% de acurácia geral quando comparado os resultados nos dois *datasets*. Todavia, é importante enfatizar o processo manual de verificação e anotação que foi feito pelos autores do *dataset* LIAR, além da documentação indicar onde foram coletados os dados, consequentemente contribuindo para a qualidade e autenticidade da base de dados. O *dataset* também mostra-se interessante com a combinação de informações das colunas, que influenciam na sua precisão e são uma maneira mais inteligente de explorar a base de dados.

Para trabalhos futuros, pretende-se realizar uma otimização dos modelos de *Machine Learning*, e também aplicar técnicas de *stemming* em conjunto com a união de colunas do dataset. Adicionalmente, aumentar a quantidade de *epochs* utilizadas no modelo BERT, e observar o quanto sua precisão pode aumentar com mais tempo de treinamento, até que a estabilização do modelo. Considerando que este trabalho é uma componente de um projeto mais amplo, pondera-se a relevância de disponibilizar os conjuntos de dados e repositórios empregados neste estudo em uma fase subsequente. Essa iniciativa visa fomentar a prática da ciência aberta e facilitar a replicação deste estudo.

REFERÊNCIAS

[1] WARDLE, C.; DERAKHSHAN, H. *Information disorder: Toward an interdisciplinary framework for research and policymaking*. [S.l.]: Council of Europe Strasbourg, 2017. v. 27.

[2] NEWMAN, N. Overview and key findings of the 2022 digital news report. *Reuters Institute for the Study of Journalism. https://reutersinstitute. politics. ox. ac. uk/digital-news-report/2022/dnr-executive-summary*, 2022.

[3] BROWN, T. et al. Language models are few-shot learners. *Advances in neural information processing systems*, v. 33, p. 1877–1901, 2020.

[4] ZHU, Q.; LUO, J. Generative pre-trained transformer for design concept generation: an exploration. *Proceedings of the design society*, Cambridge University Press, v. 2, p. 1825–1834, 2022.

[5] MEGAHED, F. M. et al. How generative ai models such as chatgpt can be (mis) used in spc practice, education, and research? an exploratory study. *Quality Engineering*, Taylor & Francis, p. 1–29, 2023.

[6] MORAN, C. Chatgpt is making up fake guardian articles. here's how we're responding. *The Guardian*, v. 6, 2023.

[7] MONTORO-MONTARROSO, A. et al. Fighting disinformation with artificial intelligence: Fundamentals, advances and challenges. *Profesional de la información*, v. 32, n. 3, 2023.

[8] SHU, K. et al. Fake news detection on social media: A data mining perspective. *ACM SIGKDD explorations newsletter*, ACM New York, NY, USA, v. 19, n. 1, p. 22–36, 2017.

[9] GUO, B. et al. The future of false information detection on social media: New perspectives and trends. *ACM Computing Surveys (CSUR)*, ACM New York, NY, USA, v. 53, n. 4, p. 1–36, 2020.

[10] MEEL, P.; VISHWAKARMA, D. K. Fake news, rumor, information pollution in social media and web: A contemporary survey of state-of-the-arts, challenges and opportunities. *Expert Systems with Applications*, Elsevier, v. 153, p. 112986, 2020.

[11] ZHANG, X.; GHORBANI, A. A. An overview of online fake news: Characterization, detection, and discussion. *Information Processing & Management*, Elsevier, v. 57, n. 2, p. 102025, 2020.

[12] SOUZA, M. C. de et al. A network-based positive and unlabeled learning approach for fake news detection. *Machine learning*, Springer, v. 111, n. 10, p. 3549–3592, 2022.

[13] DONG, X.; VICTOR, U.; QIAN, L. Two-path deep semisupervised learning for timely fake news detection. *IEEE Transactions on Computational Social Systems*, IEEE, v. 7, n. 6, p. 1386–1398, 2020.

[14] LI, X. et al. A novel self-learning semi-supervised deep learning network to detect fake news on social media. *Multimedia tools and applications*, Springer, v. 81, n. 14, p. 19341–19349, 2022.

[15] MEEL, P.; VISHWAKARMA, D. K. A temporal ensembling based semi-supervised convnet for the detection of fake news articles. *Expert Systems with Applications*, Elsevier, v. 177, p. 115002, 2021.

[16] PAKA, W. S. et al. Cross-sean: A cross-stitch semi-supervised neural attention model for covid-19 fake news detection. *Applied Soft Computing*, Elsevier, v. 107, p. 107393, 2021.

[17] NØRREGAARD, J.; HORNE, B. D.; ADALI, S. Nela-gt-2018: A large multi-labelled news dataset for the study of misinformation in news articles. In: *Proceedings of the international AAAI conference on web and social media*. [S.l.: s.n.], 2019. v. 13, p. 630–638.

[18] MITRA, T.; GILBERT, E. Credbank: A large-scale social media corpus with associated credibility annotations. In: *Proceedings of the international AAAI conference on web and social media*. [S.l.: s.n.], 2015. v. 9, n. 1, p. 258–267.

[19] SANTIA, G.; WILLIAMS, J. Buzzface: A news veracity dataset with facebook user commentary and egos. In: *Proceedings of the international AAAI conference on web and social media*. [S.l.: s.n.], 2018. v. 12, n. 1, p. 531–540.

[20] SINGER-VINE, J. et al. *BuzzFeed News*. 2014. Disponível em: <https://github.com/BuzzFeedNews>.

[21] THORNE, J. et al. FEVER: a large-scale dataset for fact extraction and VERification. In: *NAACL-HLT*. [S.l.: s.n.], 2018.

[22] WANG, W. Y. "liar, liar pants on fire": A new benchmark dataset for fake news detection. *arXiv preprint arXiv:1705.00648*, 2017.

[23] MANNING, C.; SCHUTZE, H. *Foundations of statistical natural language processing*. [S.l.]: MIT press, 1999.

[24] SERRANO-GUERRERO, J. et al. Sentiment analysis: A review and comparative analysis of web services. *Information Sciences*, Elsevier, v. 311, p. 18–38, 2015.

[25] GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep learning*. [S.l.]: MIT press, 2016.

[26] BOEHMKE, B. *New Tutorial on Logistic Regression*. 2017. Disponível em: <https://uc-r.github.io/2017/02/03/logistic-regression/>.

[27] LAVALLEY, M. P. Logistic regression. *Circulation*, Am Heart Assoc, v. 117, n. 18, p. 2395–2399, 2008.

[28] KELLEHER, J.; NAMEE, B. M.; D'ARCY, A. Machine learning for predictive data analytics. *Fundamentals of machine learning for predictive data analytics: algorithms, worked examples, and case studies*, p. 1–19, 2015.

[29] MAHESH, B. Machine learning algorithms-a review. *International Journal of Science and Research (IJSR).[Internet]*, v. 9, n. 1, p. 381–386, 2020.

[30] BOEHMKE, B. *Support Vector Machines*. 2017. Disponível em: <https://uc-r.github.io/2017/09/27/support-vector-machines/>.

[31] NAGASHRI, K.; SANGEETHA, J. Fake news detection using passive-aggressive classifier and other machine learning algorithms. In: SPRINGER. *Advances in Computing and Network Communications: Proceedings of CoCoNet 2020, Volume 2*. [S.l.], 2021. p. 221–233.

[32] BOEHMKE, B. *Random Forests*. 2018. Disponível em: <https://uc-r.github.io/2018/05/09/random-forests/>.

[33] ROGERS, A.; KOVALEVA, O.; RUMSHISKY, A. A primer in bertology: What we know about how bert works. *Transactions of the Association for Computational Linguistics*, MIT Press One Rogers Street, Cambridge, MA 02142-1209, USA journals-info …, v. 8, p. 842–866, 2021.

[34] JAIN, P. *Fake News Detection*. 2020. Disponível em: <https://www.kaggle.com/datasets/jainpooja/fake-news-detection/data>.

[35] PASZKE, A. et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, v. 32, p. 8026–8037, 2019. Disponível em: <http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf>.

[36] KRAMER, O.; KRAMER, O. Scikit-learn. *Machine learning for evolution strategies*, Springer, p. 45–53, 2016.

[37] GOOGLE. *Welcome To Colaboratory - Google Research*. Disponível em: <https://colab.research.google.com/notebooks/>.