

# GRU vs LSTM: Detecção de Notícias Falsas

Diego Machado Cordeiro, Diogo Martins de Assis, Luiz Felipe Vieira,  
Marcos Paulo Freitas da Silva, Renato Paganini Thürler Filho

*Departamento de Engenharia de Software*  
*Pontifícia Universidade Católica de Minas Gerais*  
Belo Horizonte, Minas Gerais, Brasil

{diego.cordeiro, diogo.assis, marcos.silva.1349199, renato.filho.1094209}@sga.pucminas.br,  
@sga.pucminas.br (Luiz Felipe Vieira)

**Abstract**—Este estudo tem como objetivo comparar o desempenho de duas arquiteturas de redes neurais recorrentes — LSTM (Long Short-Term Memory) e GRU (Gated Recurrent Unit) — na tarefa de classificação binária de notícias como verdadeiras ou falsas. Para isso, foi utilizado o conjunto de dados ISOT Fake News Detection Dataset, composto por textos jornalísticos rotulados. O processo metodológico incluiu etapas de pré-processamento textual, tokenização, vetorização e padronização das sequências, seguidas pela construção e treinamento dos modelos com camadas de embedding, redes recorrentes bidirecionais e funções de ativação apropriadas. Avaliação foi conduzida com base em métricas como acurácia, precisão, recall, F1-score e AUC. Os resultados obtidos indicam que ambas as arquiteturas são eficazes na detecção de desinformação textual, com desempenho elevado e consistente, contribuindo para o desenvolvimento de soluções automatizadas no combate à propagação de notícias falsas

**Index Terms**—Fake news, GRU, LSTM, IOST

## I. INTRODUÇÃO

Com o crescimento exponencial do acesso à informação por meio da internet e das redes sociais, a disseminação de notícias falsas tornou-se um fenômeno preocupante, com impactos significativos na sociedade, na política e na saúde pública. A facilidade com que conteúdos enganosos são compartilhados compromete a credibilidade de fontes legítimas e dificulta o acesso a informações confiáveis. Nesse contexto, torna-se essencial o desenvolvimento e a avaliação de métodos automáticos capazes de identificar e filtrar conteúdos falsos de forma eficiente. [1]

A detecção automática de notícias falsas é um problema recorrente de classificação de texto, no qual o objetivo é atribuir uma categoria (neste caso, "verdadeira" ou "falsa") a um determinado conteúdo textual. Esse tipo de tarefa pode ser abordado por meio de técnicas de aprendizado de máquina, que consistem em treinar algoritmos para reconhecer padrões nos dados e fazer previsões com base nesses padrões. [2]

Nos últimos anos, os avanços em aprendizado profundo (deep learning) — uma subárea do aprendizado de máquina — têm proporcionado resultados expressivos em tarefas de Processamento de Linguagem Natural (PLN). O aprendizado profundo utiliza redes neurais com múltiplas camadas para extrair representações complexas dos dados, sendo especialmente eficaz em tarefas que envolvem linguagem, imagem e som. [2]

Entre os modelos mais promissores para o tratamento de sequências de texto estão as redes neurais recorrentes (Recurrent Neural Networks – RNNs). Diferentemente das redes neurais tradicionais, as RNNs são projetadas para lidar com dados sequenciais, como frases e documentos, pois possuem uma estrutura que permite "memorizar" informações anteriores da sequência.

Duas variantes avançadas das RNNs são as arquiteturas LSTM (Long Short-Term Memory) e GRU (Gated Recurrent Unit). Ambas foram desenvolvidas para resolver limitações das RNNs tradicionais, como o problema do desvanecimento do gradiente, que dificulta o aprendizado de dependências de longo prazo em sequências.

A LSTM introduz mecanismos chamados de portas (gates), que controlam o fluxo de informações dentro da célula da rede. Essas portas decidem o que deve ser armazenado, esquecido ou passado adiante, permitindo que o modelo retenha informações relevantes por períodos mais longos. [3]

A GRU, por sua vez, é uma versão mais simplificada da LSTM. Ela também utiliza portas para controlar o fluxo de informação, mas com uma estrutura mais compacta e menos parâmetros, o que pode resultar em um treinamento mais rápido e eficiente, especialmente em conjuntos de dados menores ou menos complexos.

Ambas as arquiteturas são capazes de capturar dependências temporais e contextuais em sequências de texto, ou seja, conseguem entender como o significado de uma palavra pode depender das palavras anteriores e posteriores. Essa capacidade é essencial para tarefas como a detecção de fake news, onde o contexto e a forma como as informações são apresentadas podem ser determinantes para a classificação correta. [4]

## II. MÉTODO

A metodologia adotada neste estudo foi estruturada em quatro etapas principais: pré-processamento dos dados, tokenização e vetorização, divisão dos dados e construção e treinamento dos modelos. Cada uma dessas etapas é descrita a seguir, com a devida contextualização teórica.

**3.1 Pré-processamento dos Dados** O conjunto de dados utilizado foi o ISOT Fake News Detection Dataset, composto por textos de notícias rotuladas como verdadeiras ou falsas. Inicialmente, os dados foram organizados em um único conjunto, com a atribuição de rótulos binários: 1 para notícias

verdadeiras e 0 para falsas. Em seguida, os textos passaram por um processo de limpeza, no qual foram removidos caracteres especiais, números e pontuações, mantendo-se apenas letras e espaços. Além disso, todo o conteúdo textual foi convertido para letras minúsculas, com o objetivo de padronizar o vocabulário e reduzir a dimensionalidade dos dados.

Essa etapa de pré-processamento é essencial em tarefas de Processamento de Linguagem Natural (PLN), pois melhora a qualidade dos dados e facilita o aprendizado dos modelos.

**3.2 Tokenização e Vetorização** Após a limpeza textual, os dados foram transformados em uma representação numérica por meio da tokenização, processo que consiste em dividir o texto em unidades menores chamadas tokens (geralmente palavras) e mapear cada token para um número inteiro. Utilizou-se o Tokenizer da biblioteca Keras, com um limite de 5.000 palavras mais frequentes e um token especial para palavras fora do vocabulário (Out Of Vocabulary –  $\text{OOV}_i$ ).

Em seguida, foi aplicada a vetorização, que converte as sequências de tokens em vetores de comprimento fixo. Para isso, utilizou-se a técnica de padding, que preenche as sequências com zeros até que todas tenham o mesmo tamanho (neste caso, 200 tokens). Essa padronização é necessária para que os dados possam ser processados por redes neurais.

**3.3 Divisão dos Dados** O conjunto vetorizado foi dividido em dois subconjuntos: 60% dos dados foram utilizados para treinamento e 40% para teste. Além disso, durante o treinamento, 20% do conjunto de treino foi reservado para validação. Essa divisão permite avaliar o desempenho do modelo em dados não vistos, garantindo uma estimativa mais realista de sua capacidade de generalização.

**3.4 Construção e Treinamento do Modelo** A arquitetura do modelo foi composta pelas seguintes camadas:

**Camada de Embedding:** transforma cada token em um vetor denso de 64 dimensões. Essa camada aprende representações semânticas das palavras durante o treinamento.

**Camada Bidirecional com GRU:** a Gated Recurrent Unit (GRU) é um tipo de rede neural recorrente (RNN) eficiente para modelar sequências. A versão bidirecional permite que o modelo processe o texto em ambas as direções, capturando melhor o contexto.

**Camada de Dropout:** com taxa de 50%, desativa aleatoriamente neurônios durante o treinamento, reduzindo o risco de overfitting.

**Camada Densa com ReLU:** uma camada totalmente conectada com 32 unidades e função de ativação ReLU (Rectified Linear Unit), que introduz não linearidade ao modelo. A função ReLU retorna zero para valores negativos e o próprio valor para positivos, acelerando o treinamento.

**Camada de Saída com Sigmoid:** a função de ativação sigmoide transforma a saída do modelo em um valor entre 0 e 1, interpretado como a probabilidade de a notícia ser verdadeira. Essa função é definida como:

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

Por ser uma função contínua e diferenciável, a sigmoide é amplamente utilizada em problemas de classificação binária.

O modelo foi compilado com a função de perda binary crossentropy, apropriada para tarefas de classificação binária, e o otimizador Adam, que ajusta os pesos da rede de forma adaptativa. As métricas utilizadas para avaliação foram a acurácia e a AUC (Área sob a Curva ROC), que mede a capacidade do modelo de distinguir entre as classes.

Durante o treinamento, foram utilizados dois callbacks:

**EarlyStopping:** interrompe o treinamento caso a métrica de validação pare de melhorar, evitando o sobreajuste. **ReduceLROnPlateau:** reduz automaticamente a taxa de aprendizado quando o desempenho do modelo se estabiliza, facilitando a convergência.

### III. RESULTADOS

#### A. Conjunto de Dados

O conjunto de dados utilizado neste estudo foi o *ISOT Fake News Detection Dataset*, amplamente empregado em pesquisas sobre detecção de desinformação. Ele é composto por dois subconjuntos: um contendo notícias verdadeiras, extraídas de fontes confiáveis como a Reuters, e outro com notícias falsas, provenientes de websites conhecidos por disseminar conteúdo enganoso. Para este trabalho, foi considerada apenas a coluna `text`, que contém o corpo da notícia.

Após a junção dos subconjuntos e a rotulagem binária (1 para notícias verdadeiras e 0 para falsas), o corpus totalizou 44.898 amostras. Os dados foram embaralhados e divididos em cinco subconjuntos para validação cruzada (*cross-validation*), garantindo uma avaliação mais robusta e generalizável dos modelos.

#### B. Resultados com a Arquitetura GRU

O modelo baseado na arquitetura GRU (*Gated Recurrent Unit*) foi treinado utilizando validação cruzada com cinco dobras (*folds*), cada uma com cinco épocas de treinamento. A Tabela I resume os principais resultados obtidos em cada dobra:

TABLE I  
RESULTADOS POR DOBRA PARA O MODELO GRU

Dobra	Acurácia	Precisão	Recall	F1-score
Fold 1	98,85%	0,99	0,99	0,99
Fold 2	98,00%	0,99	0,97	0,98
Fold 3	98,00%	0,99	0,98	0,98
Fold 4	98,00%	0,98	0,98	0,98
Fold 5	98,00%	0,99	0,98	0,98

A matriz de confusão revelou uma taxa de erro extremamente baixa, com equilíbrio entre falsos positivos e falsos negativos. As curvas de perda e AUC ao longo das épocas indicaram uma boa convergência do modelo, sem sinais de sobreajuste, mesmo com o aumento da acurácia nas últimas épocas.

Esses resultados demonstram que a arquitetura GRU é altamente eficaz na tarefa de classificação de notícias falsas,

apresentando desempenho consistente e robusto em diferentes partições dos dados.

### C. Resultados com a Arquitetura LSTM

O modelo baseado na arquitetura *LSTM* (Long Short-Term Memory) foi treinado com os mesmos parâmetros utilizados no modelo GRU, incluindo divisão dos dados, número de épocas, tamanho do lote e uso de *callbacks* para parada antecipada e ajuste da taxa de aprendizado. O treinamento foi realizado por cinco épocas, com validação em 20% do conjunto de treino.

Durante o treinamento, o modelo apresentou melhora progressiva nas métricas de desempenho. A acurácia no conjunto de validação atingiu até 98,50%, com valores de perda decrescentes e estáveis. A Tabela II apresenta os principais resultados obtidos em duas execuções representativas.

TABLE II  
RESULTADOS POR DOBRA PARA O MODELO LSTM

Dobra	Acurácia	Precisão	Recall	F1-score
Fold 1	98,69%	0,98	0,99	0,99
Fold 2	98,00%	0,99	0,98	0,98

No conjunto de teste final, com 17.960 amostras, o modelo LSTM apresentou os seguintes resultados:

- **Acurácia:** 98,56%
- **AUC:** 0,9974
- **Precisão (Fake):** 0,98
- **Recall (Fake):** 0,99
- **F1-score (Fake):** 0,99
- **Precisão (True):** 0,99
- **Recall (True):** 0,98
- **F1-score (True):** 0,98

A matriz de confusão revelou um desempenho equilibrado entre as classes, com baixas taxas de falsos positivos e falsos negativos. As curvas de perda e AUC ao longo das épocas indicaram uma boa convergência do modelo, com estabilidade nas últimas iterações.

Esses resultados demonstram que a arquitetura LSTM também é altamente eficaz na tarefa de classificação de notícias falsas, com desempenho comparável ao modelo GRU.

### D. Comparação entre os Modelos GRU e LSTM

Ambas as arquiteturas, GRU e LSTM, apresentaram desempenho elevado na tarefa de classificação binária de notícias falsas e verdadeiras. No entanto, algumas diferenças sutis foram observadas em termos de desempenho e comportamento durante o treinamento.

O modelo GRU obteve uma acurácia média ligeiramente superior nas dobras de validação cruzada, com destaque para a Dobra 1, que alcançou 98,85% de acurácia e F1-score de 0,99. Já o modelo LSTM apresentou acurácia de até 98,69% em uma das execuções, com desempenho muito próximo ao do GRU.

No conjunto de teste final, os dois modelos apresentaram resultados praticamente equivalentes. O GRU obteve uma

acurácia de 98,92% e AUC de 0,9988, enquanto o LSTM alcançou 98,56% de acurácia e AUC de 0,9974. Em termos de métricas por classe, ambos os modelos demonstraram equilíbrio entre precisão e recall para as classes *Fake* e *True*, com F1-scores de 0,98 ou 0,99 em todos os casos.

Do ponto de vista computacional, o GRU tende a ser mais leve e rápido para treinar, devido à sua estrutura mais simples em comparação com o LSTM. Isso pode representar uma vantagem em cenários com restrições de tempo ou recursos computacionais.

Em resumo, os dois modelos mostraram-se altamente eficazes na detecção de desinformação textual. A escolha entre GRU e LSTM pode depender de fatores como tempo de treinamento, disponibilidade de recursos e preferência por simplicidade ou capacidade de modelagem de dependências mais longas.

A análise realizada neste trabalho evidenciou a viabilidade do uso de redes neurais recorrentes na tarefa de classificação de notícias quanto à sua veracidade. A partir da aplicação de duas arquiteturas distintas, foi possível observar que modelos baseados em sequências textuais são capazes de capturar padrões linguísticos relevantes para a detecção de desinformação.

## IV. CONCLUSÕES

Os experimentos realizados demonstraram que tanto as redes GRU quanto LSTM alcançaram resultados expressivos na tarefa de detecção de fake news, com métricas de desempenho elevadas e consistentes ao longo das diferentes dobras de validação e no conjunto de teste. Em particular, o modelo GRU apresentou uma leve vantagem em termos de acurácia geral (98,92% contra 98,56% da LSTM), enquanto ambos os modelos obtiveram F1-scores muito próximos, indicando desempenho equilibrado entre precisão e recall.

Esses achados reforçam a robustez de ambas as abordagens, mas também destacam a importância de considerar o contexto de aplicação. A GRU, por sua arquitetura mais enxuta, é particularmente vantajosa em cenários com restrições de tempo ou infraestrutura computacional limitada. Já a LSTM se mostra mais adequada em aplicações que exigem maior profundidade na modelagem sequencial, como em textos mais longos ou com dependências temporais mais complexas.

Portanto, a escolha entre GRU e LSTM deve ser orientada não apenas pelos resultados quantitativos, mas também por fatores práticos e estratégicos, como os objetivos do sistema, os recursos disponíveis e a natureza dos dados. Essa análise comparativa contribui para uma tomada de decisão mais informada e alinhada às necessidades reais do problema.

## REFERENCES

- [1] N. Chaudhuri, G. Gupta, M. Bagherzadeh, T. Daim, and H. Yalcin, "Misinformation on social platforms: A review and research agenda," *\*Technology in Society\**, vol. 78, p. 102654, 2024. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0160791X24002021>
- [2] S. Khare, P. Singh, and P. Kumar, "An analysis of various classification algorithms for fake news detection," in *\*Proc. 2025 2nd Int. Conf. Comput. Intell., Commun. Technol. Netw. (CICTN)\**, 2025, pp. 268–272, doi: 10.1109/CICTN64563.2025.10932522

- [3] T. S. Camelia, F. R. Fahim, and M. M. Anwar, "A regularized LSTM method for detecting fake news articles," in \*Proc. 2024 IEEE Int. Conf. Signal Process., Inf., Commun. Syst. (SPICSCON)\*, 2024, pp. 01–06, doi: 10.1109/SPICSCON64195.2024.10941441.
- [4] S. R. Tanuku, "Novel approach to capture fake news classification using LSTM and GRU networks," in \*Proc. 2022 Int. Conf. Futur. Technol. (INCOFT)\*, Belgaum, India, 2022, pp. 1–4, doi: 10.1109/INCOFT55651.2022.10094467