**Fake News Detection with Machine Learning and Deep Learning Models**

Ruddy Simonpour

Arya Shahbazi

Mohammad Mahmoudighaznavi

University of San Diego

Shiley-Marcos School of Engineering

ADS 504 - Machine Learning

<https://github.com/ruddysimon/ADS504-machine-learning-deep-learning>

August 2023

**Abstract**

The rise of fake news dissemination has triggered concerns. It can mislead the public, undermine trust in the media, deepen societal divides, and threaten democratic principles. This research introduces a methodology for detecting fake news using machine learning and deep learning techniques. Using a dataset of 20,800 news articles from Kaggle, we evaluated and compared the performance of several models, including Logistic Regression, Decision Trees, Support Vector Machines, Long Short-Term Memory (LSTM), Recurrent Neural Networks (RNN), and Gated Recurrent Unit (GRU) networks. Traditional machine learning models consistently outperformed deep learning approaches, with Logistic Regression and SVM achieving test accuracies exceeding 96%. This study also identifies potential avenues for enhancement, emphasizing data diversity, improved handling of missing values, regular updates, diverse feature incorporation, exploration of alternative models, and model interpretability. Our findings offer crucial insights into the intricacies and prospects of automated fake news detection, aiming to foster a more accurate and reliable information ecosystem.

*Keywords:* Deep learning, fake news detection, data diversity, machine learning, Logistic regression, Support Vector Machines, LSTM, RNN, GRU, model performance, model interpretability.

# Table of contents

Table of Contents

[Table of Contents 3](#_Toc142504229)

[Introduction 4](#_Toc142504230)

[Problem Statement 4](#_Toc142504231)

[Methodology 4](#_Toc142504232)

[GitHub Repository Link 5](#_Toc142504233)

[Source Datasets 5](#_Toc142504234)

[Data Exploration 5](#_Toc142504235)

[Data Pre-Processing 8](#_Toc142504236)

[Modeling 10](#_Toc142504237)

[Model Evaluations 11](#_Toc142504238)

[Comparative Evaluation 12](#_Toc142504239)

[Feature Enhancements 13](#_Toc142504240)

[Conclusion 15](#_Toc142504241)

## Introduction

Deep learning has dominated the landscape of various machine learning tasks in recent years, particularly those involving large-scale, high-dimensional data, such as picture and voice recognition. Detecting and classifying fake news presents unique challenges beyond typical data identification. Fake news can drastically alter public perception, undermine trust in authentic news sources, and even endanger democratic processes. Despite deep learning's success in many areas, its role in fake news detection still needs to be debated. This study compares classical machine learning approaches to deep learning methods to determine their effectiveness in detecting fake news and building a more accurate and reliable information ecosystem.

## Problem Statement

Fake news presents misleading information as factual and is a global challenge with wide-reaching implications (Allcott & Gentzkow, 2017). Such misinformation not only misleads the public but also reinforces biases (Wardle & Derakhshan, 2017), erodes trust in traditional media (Tandoc Jr et al., 2018), deepens societal divisions, and undermines democratic processes with substantial social and economic consequences. Machine learning is pivotal in fighting (Shu et al., 2017).

## ****Methodology****

**We developed a methodology tailored for fake news detection using machine learning and deep learning. We used a Kaggle dataset comprising 20,800 news articles labeled as fake or real. After briefly analyzing the dataset's attributes and target variable distribution, we proceeded with data preprocessing. This involved removing extraneous characters using regular expressions, tokenizing, lemmatizing, removing stopwords, and converting text to numerical format with the TF-IDF vectorizer.**

**We tested a spectrum of models for development, from traditional ones like Logistic Regression and Decision Trees to advanced deep learning models such as LSTM, RNN, and GRU. All models were tuned for best performance, with efficacy gauged via AUC, training, and test accuracies.**

**Our research culminated in a comparative assessment of traditional versus deep learning models for fake news detection, followed by recommendations for enhancing future studies, emphasizing data diversity and model clarity.**

### GitHub Repository Link

Our GitHub repository can be found at: <https://github.com/ruddysimon/ADS504-machine-learning-deep-learning>. It includes a dataset and all the codes in python to reproduce the outcomes.

### Source Datasets

We are looking at open-source fake news data from Kaggle <https://www.kaggle.com/datasets/ronikdedhia/fake-news>. This dataset contains 20800 observations, and each row has five measurements, including the label for news as our target variable. Number 1 in the label column stands for fake news, and number 0 shows the real news.

### Data Exploration

This dataset contains information of 5 following columns:

* id: An identifier for each news article.
* title: The title of the news article.
* author: The author of the news article.
* text: The main content or body of the news article.
* label: A binary label indicating whether the news is fake (1) or real (0).

This dataset is missing some values. The breakdown of missing values for each column has shown in Table 1.

Table 1:

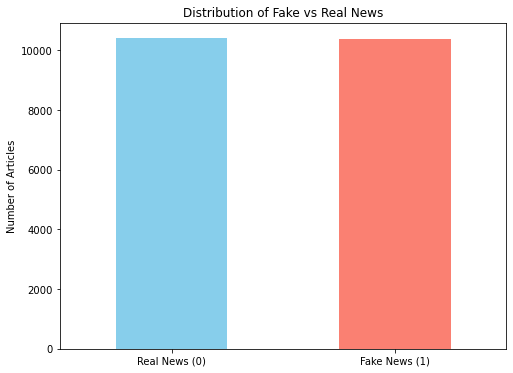
*Missing Values Report*

|  |  |
| --- | --- |
| Column Name | Number of missing values |
| id | 0 |
| title | 558 |
| author | 1957 |
| text | 39 |
| label | 0 |

To create a fair model, we must explore the target variable's distribution (label). Figure 1 shows the distribution of the label column in our dataset. The dataset appears to be balanced; both real news (label 0) and fake news (label 1) have approximately the same number of articles.

Figure 1:

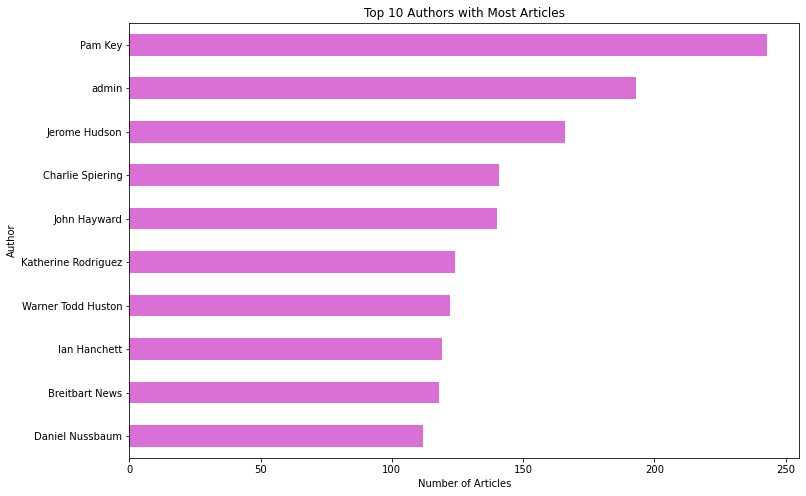
*Distribution of Target Variable*

****

The top 10 authors with the most articles in the dataset are shown in Figure 2. This figure shows that some authors have contributed significantly more articles than others.

Figure 2:

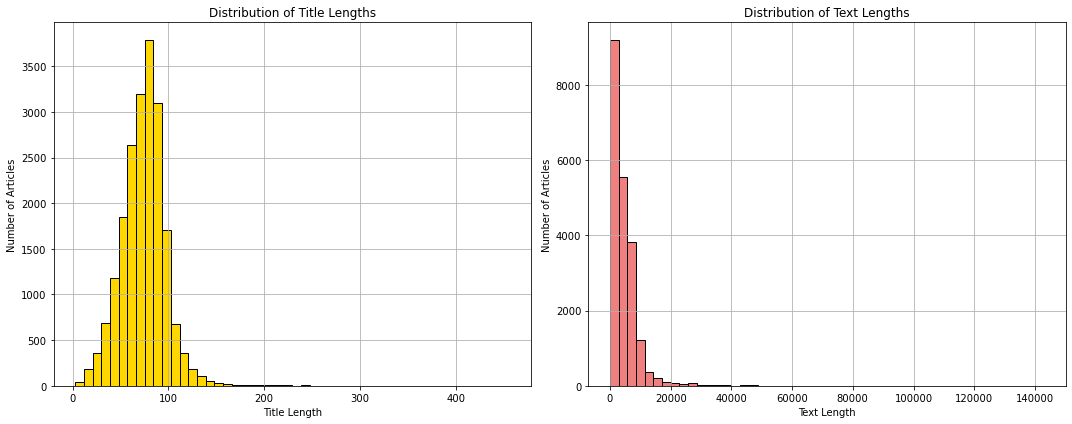
*Top 10 Authors with the Most Articles*

**

The distribution of article length will give us insights into the nature of the articles and whether there are significant outliers in length. As shown in Figure 3, most article titles have lengths ranging from 40 to 80 characters, and only some titles are extremely short or long. Most articles have text lengths of up to 5000 characters in the text column.

Figure 3:

*Distribution of Title and Text Lengths*

 **Data Pre-Processing**

During the preparation stage of our project, we worked hard to prepare the text data for machine learning models. Before using the text data in a model, we realized it needed considerable cleaning and processing.

We removed any unwanted letters, symbols, or spaces from the data. Regular expressions were used to replace everything except letters and spaces. This technique ensured that our dataset contained only important textual information and significantly reduced noise.

After cleaning the text, we tokenized it, breaking it into individual words or tokens. We chose this strategy because machine learning models do not perceive text like people do; instead, they require numerical or categorical input. Tokenization was the first step in converting our text input into a format our models could recognize.

Lemmatization, a technique used to reduce words to their base or root form, was also added to our preparation workflow (Bird et al., 2009). Lemmatization is a technique used in computer linguistics and natural language processing (NLP), including chatbot development, to aggregate numerous inflected word variants. This method simplifies the grouping of many word forms, allowing them to be studied as a single root word (Gillis, 2023). For example, the words 'running,' 'runs,' and 'ran' would all be shortened to 'run.' This phase is critical in NLP projects because it allows the model to handle diverse inflections of a word as a single item by grouping them.

We then eliminated any stop words from the data. Stop words have no meaningful meaning and are typically eliminated from texts. They include phrases like 'is,' 'an,' and 'the.' The NLTK library was utilized, which offers a list of commonly agreed-upon stop words. Finally, after completing all the preceding processes, we vectorized the text. Text vectorization converts text into a format machine learning algorithms can understand. For this, we employed the TF-IDF vectorizer. The Term Frequency-Inverse Document Frequency (TF-IDF) statistic measures the relevance of a word about a document in a corpus (Géron, 2022). Using this strategy, we could translate our text data into a comprehensible representation without sacrificing any vital information.

We had a clean, standardized dataset ready for model creation by the end of our preprocessing. We did not introduce bias or unnecessary noise into the data during preparation. Each step was critical to the success of our later machine-learning models.

## Modeling

To tackle the problem of fake news categorization, we utilized an array of linear and non-linear models, incorporating the benefits of traditional machine learning algorithms and cutting-edge deep learning methods. Among the traditional models, we employed Logistic Regression, Decision Trees, and Support Vector Machines (SVM) with a Linear Kernel. These models offer diverse strengths: Logistic Regression for its simplicity and interpretability, Decision Trees for their non-linear decision boundaries and intuitive understanding, and SVM for its robustness in handling complex datasets.

For deep learning, we utilized LSTM, RNN, and GRU networks. These models are designed to analyze sequential data and capture complex patterns, making them ideally suited for text data.

We fine-tuned hyperparameters for each model to optimize performance and prevent underfitting or overfitting. We used techniques like RandomizedSearchCV for traditional models and adjusted the number of epochs and batch size for deep learning models.

We evaluated our models using the Area Under the Receiver Operating Characteristic Curve (AUC) statistic, which considers both false positive and false negative rates. This and balancing the dataset during preprocessing helped mitigate potential difficulties associated with imbalanced data.

Figure 4:

*ROC Plot*

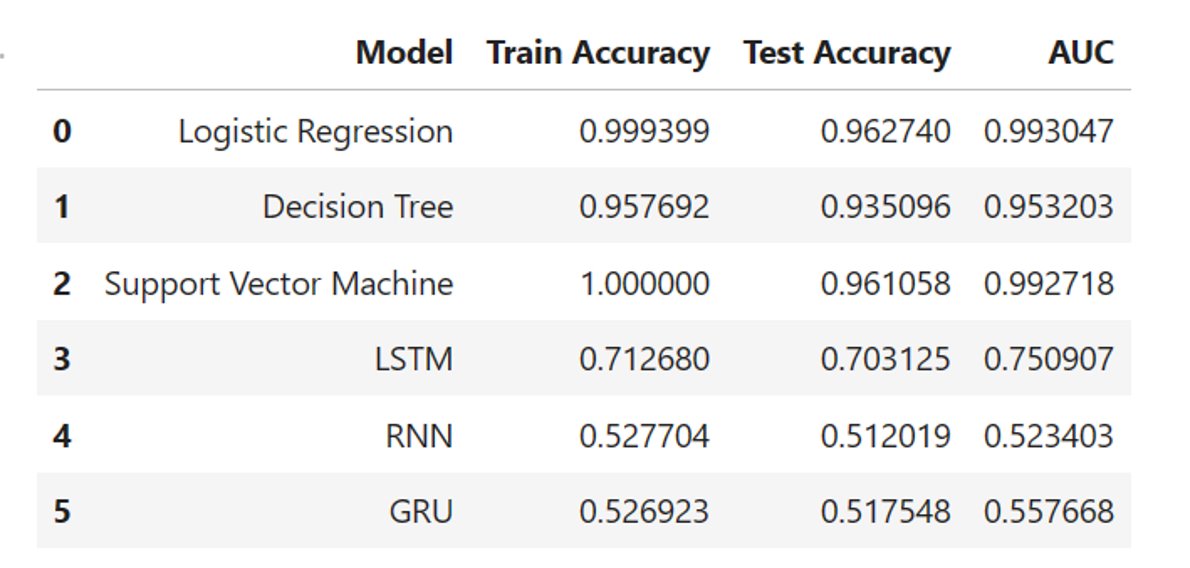
A graph of a logistic plot

Description automatically generated

## Model Evaluations

Figure 5:

*Model Evaluation Table*



We employed a variety of models in our analysis:

* LSTM: Achieved a test accuracy of 70.31% after ten epochs.
* RNN: Had consistent validation accuracy of roughly 51.20% across ten epochs, suggesting potential underfitting.
* GRU: Also showed potential underfitting, with a consistent validation accuracy of around 51% over ten epochs.
* Logistic Regression: Achieved a test accuracy of 96.27%.
* Decision Tree: Attained a test accuracy of 93.51%.
* SVM: Scored a test accuracy of 96.11%.

We inferred overfitting and underfitting trends from the discrepancy between training and testing accuracy for the traditional m. During training, we relied on cross-validation scores to avoid overfitting and underfitting.

Based on the results of the evaluations, we selected Logistic Regression as the final model due to its high test accuracy of 96.27% and low computing cost, making it a viable solution for our current needs.

## Comparative Evaluation

When comparing the performance of the models, traditional machine learning methods outperformed the deep learning models. Logistic Regression and SVM models achieved test accuracies over 96%, while the Decision Tree model achieved an accuracy of 93.51%. In contrast, the deep learning models (LSTM, RNN, GRU) had 51.20% and 70.31% accuracy.

Although the traditional models had nearly perfect training accuracies, their high-test accuracies indicate they generalized effectively to unseen data rather than overfitting. In contrast, the deep learning models had significantly lower training and test accuracies, suggesting they may be underfitting the data. Adjusting the deep learning models' architecture or training procedure might improve their performance. However, this also brings the risk of overfitting and thus requires careful balancing. Given the continuous advancements in deep learning, we prefer investing more in these models for future tasks.

The relatively lower performance of the deep learning models highlights the potential to explore other model types or even a combination of models (an ensemble). For instance, Convolutional Neural Networks (CNNs) or transformer-based models may improve performance on this task.

Our results emphasize the importance of model selection and the potential of combining different model types. Future work could explore other models, such as CNNs or transformer-based models, or investigate ensemble methods.

## Feature Enhancements

Our project has significantly progressed in modeling and feature engineering for fake news classification. Nevertheless, there is potential for further refinement. We suggest the following enhancements:

* **Data Augmentation:** Broaden our dataset to cover diverse fake news scenarios.
* **Multimodal Models:** Combine text, images, and audio for richer fake news detection.
* **Transfer Learning:** Utilize pre-trained models for better accuracy and faster training.
* **Explainability:** Make our model's decisions transparent and interpretable.
* **Feedback Mechanisms:** Allow user feedback for continuous model improvement.
* **Real-time Analysis:** Focus on immediate fake news detection for timely intervention.
* **Cross-language Capabilities:** Aim for a global reach by incorporating multiple languages.

**Data Diversity**: Our dataset comprises 20,800 news articles. However, they originate from various online news platforms, which could inherently contain biases related to political leaning, geographic location, or the type of news covered. These biases may affect the model's performance when applied to more diverse or different news articles. Future studies should consider collecting data from a broader range of sources.

**Handling Missing Values**: Our dataset has some missing values, particularly in the 'title' (558 lost entries), 'author' (1957 missing entries), and 'text' (39 missing entries) columns. Developing more sophisticated strategies for handling these missing values could improve our model's performance.

**Constant Model Updates**: Machine learning models are not static. As new data becomes available, models should be retrained and updated to ensure they adapt to changes in news reporting, writing styles, and the evolving nature of disinformation.

**Feature Expansion**: While our models relied heavily on text-based features, other potential features can be incorporated. These might include meta-information about the articles (such as the publication date, author, or source credibility), sentiment analysis results, or even features extracted from user comments or reactions.

**Exploring Other Models or Techniques**: Our results suggest that traditional machine learning models outperformed deep learning models in this project. However, other sophisticated models and techniques, such as Convolutional Neural Networks (CNNs) or transformer-based models like BERT, could yield better results.

**Model Interpretability**: Understanding why a model makes specific predictions is equally essential as achieving high accuracy, especially in a sensitive area like fake news detection. Future work could focus on enhancing model interpretability, possibly through the use of explainable AI (XAI) techniques or by incorporating more interpretable models into our ensemble.

Our goal is to enhance our system’s reliability, making it a vital tool against disinformation.

## Conclusion

In the digital age, the battle against disinformation is more crucial than ever. Our unwavering commitment to this cause drives us forward. Our goal remains clear as we continue to refine our system: to establish it as a vital tool against disinformation.

**Reference**

Allcott, H., & Gentzkow, M. (2017). Social media and fake news in the 2016 election. Journal of Economic Perspectives, 31(2), 211-236.

Bird, S., Klein, E., & Loper, E. (2009). Natural language processing with Python: analyzing text with the natural language toolkit. O'Reilly Media, Inc.

Géron, A. (2022). Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems. O'Reilly Media, Inc.

Gillis, A. S. (2023). lemmatization. Enterprise AI. <https://www.techtarget.com/searchenterpriseai/definition/lemmatization?Offer=abt_pubpro_AI-Insider>

GeeksforGeeks. (2023). ML Underfitting and overfitting. GeeksforGeeks. <https://www.geeksforgeeks.org/underfitting-and-overfitting-in-machine-learning/>

Shu, K., Sliva, A., Wang, S., Tang, J., & Liu, H. (2017). Fake news detection on social media: A data mining perspective. ACM SIGKDD Explorations Newsletter, 19(1), 22-36.

Tandoc Jr, E. C., Lim, Z. W., & Ling, R. (2018). Defining “fake news.” Digital journalism, 6(2), 137-153.

Wardle, C., & Derakhshan, H. (2017). Information disorder: Toward an interdisciplinary framework for research and policymaking. Council of Europe report, 27(2017), 1-93.