Fake News Detection

Project proposal 4

1st Artur Almeida 123196
Department of electronics, telecommunications and Informatics
University of Aveiro
Aprendizagem Automática
Instructor: Petia Georgieva
arturalmeida@ua.pt

2nd Rafael Morgado 104277
Department of electronics, telecommunications and Informatics
University of Aveiro
Aprendizagem Automática
Instructor: Petia Georgieva
rafa.morgado@ua.pt

***Abstract*—The proliferation of fake news on digital platforms has raised significant concerns about information integrity and public trust. This project explores the application of machine learning and deep learning methods to the problem of fake news detection using the ”Fake and Real News” dataset from Kaggle. We perform extensive data preprocessing, including text cleaning and removal of news agency prefixes, followed by exploratory data analysis to identify relevant patterns and biases. Two types of models are implemented and compared: classical machine learning models such as Logistic Regression and Support Vector Machines using TF-IDF features, and a transformer-based deep learning model (FakeBERT), which combines BERT embeddings with a convolutional neural network. Experimental results show that while classical models provide high performance with lower computational cost, the FakeBERT model achieves superior accuracy ($\approx 99\%$), F1-score, and AUC, demonstrating its effectiveness in capturing complex semantic patterns. We conclude with a discussion of challenges in generalizing fake news detectors to unseen events and suggest directions for future work in enhancing robustness and explainability.**

***Index Terms*—Fake news, BERT, machine learning, deep learning, text classification, CNN, natural language processing**

I. Introduction

The exponential growth of social media and online news platforms has drastically transformed how individuals consume information. While this democratization of news distribution has its benefits, it also facilitates the rapid spread of misinformation, commonly referred to as ”fake news.” These fake news articles often mimic legitimate journalism, making them challenging to identify without automated tools.

Fake news can significantly influence public opinion, distort democratic processes, and generate societal confusion. During critical events such as elections or pandemics, the propagation of false information can have real-world consequences. As such, developing robust, automated systems for detecting fake news is an essential task for modern information systems.

Machine learning (ML) and natural language processing (NLP) have emerged as key technologies for addressing this challenge. Classical ML methods like Naive Bayes, Logistic Regression, and Support Vector Machines offer interpretability and efficiency. In contrast, deep learning models, including Long Short-Term Memory networks (LSTMs), Convolutional Neural Networks (CNNs), and transformer-based models like BERT, provide enhanced accuracy by capturing complex language patterns.

This report presents a comprehensive study of these methods applied to the Fake and Real News dataset. We aim to assess how different preprocessing techniques, feature representations, and model types affect the classification accuracy. Furthermore, we discuss the challenges involved in generalizing fake news detectors to unseen events and propose potential solutions inspired by recent advances in adversarial and explainable AI.

The rest of the report is organized as follows: Section II reviews related work in fake news detection. Section III describes the dataset and presents a statistical analysis. Section IV outlines preprocessing steps. Section V details the ML models and experimental setup. Section VI presents and discusses results. Finally, Section VII concludes the study and proposes future directions.

II. State of the Art

Fake news detection is not only a technically complex task because of the nuances in natural language, but also a socially important one. The fast spread of misinformation on social media and news platforms has led to the development of many approaches to automatically identify fake content.

At first, most approaches relied on basic machine learning models like Naive Bayes, Support Vector Machines, or Logistic Regression, usually combined with text representations such as TF-IDF or Bag-of-Words [4]. These models were fast and simple to implement, but they could not fully understand the meaning behind the words or the context in which they were used.

Later, deep learning models such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), especially Long Short-Term Memory (LSTM) networks, were introduced to better capture the structure and flow of text [3]. These models improved performance, but they required more data and computational resources.

Currently, the most effective models for fake news detection are transformer-based models, especially BERT (Bidirectional Encoder Representations from Transformers) [1]. BERT uses attention mechanisms to understand the meaning of words in context. FakeBERT is a variation of BERT specifically fine-tuned for fake news detection, and it has achieved higher accuracy on several benchmark datasets [5].

Several ensemble approaches have also been proposed. For instance, Ahmad et al. [2] tested SVM, LR, and other models, achieving an accuracy of up to 92% with logistic regression and SVM on social media data.

Although these models perform very well, they are often computationally expensive and may be sensitive to biases in the training data. For this reason, recent work also explores areas like model explainability, fairness, and robustness to adversarial examples.

We chose this topic not only because of its social importance, but also because we are interested in the combination of Natural Language Processing and deep learning. We were especially curious to explore transformer-based models like BERT and apply them to a real-world classification problem. Working on fake news detection allowed us to learn more about modern machine learning techniques and the challenges of dealing with complex text data.

III. Dataset and Exploratory Analysis

The dataset used in this project is the ”fake-and-real-news-dataset”, available on Kaggle. It consists of two separate CSV files: one containing fake news articles and another with real news. Each entry includes attributes such as title, text, subject, and publication date. We merged the two files into a single dataset and added a new column called `label`, where 0 represents fake news and 1 represents real news.

After merging, we removed the duplicates and obtained a total of 38,646 news articles: 21,191 real and 17,455 fake. As shown in Figure 1, the dataset is reasonably balanced, which is important to prevent bias during model training.

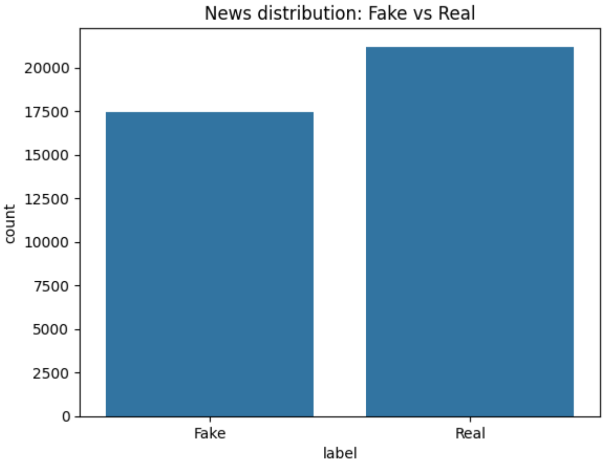

Fig. 1. Distribution of Fake and Real news articles.

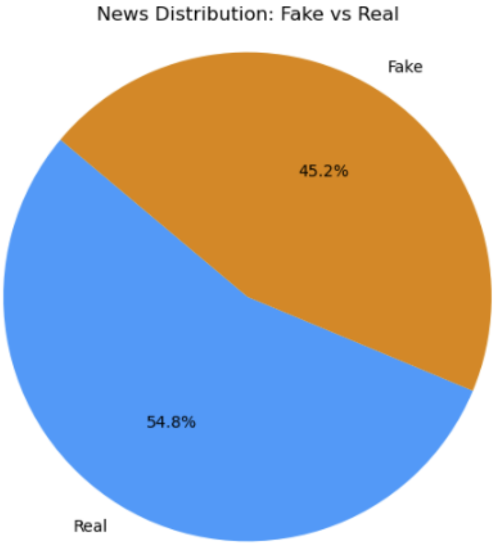

Fig. 2. Distribution of Fake and Real news articles.

We then analyzed the length of the articles by counting the number of words per text. As seen in Figure 3, real news articles are generally longer, which may suggest a higher degree of detail or completeness.

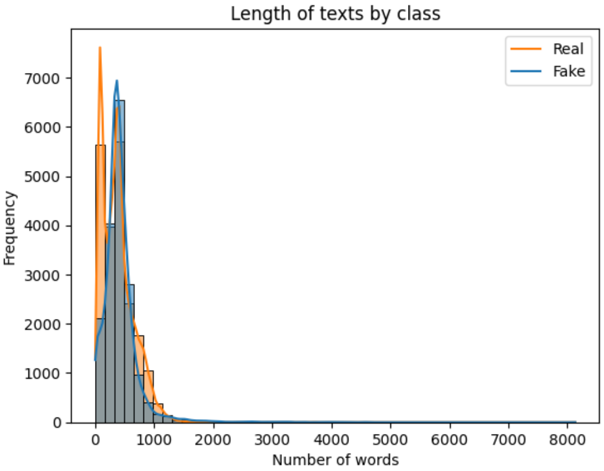

Fig. 3. Text length distribution by class. Real articles tend to be longer.

Next, we created wordclouds to visualize the most frequent words in fake and real news (Figures 4 and 5). While both mention similar political figures, fake news often includes more emotionally charged or vague terms.

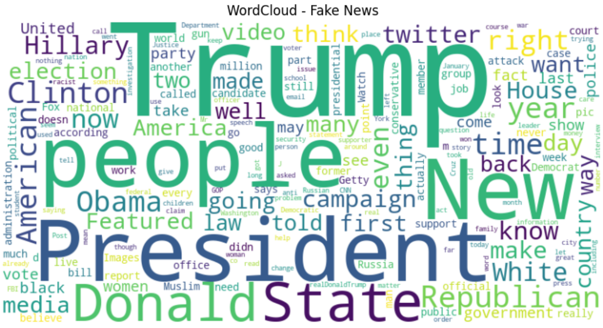

Fig. 4. Wordcloud for Fake news articles.

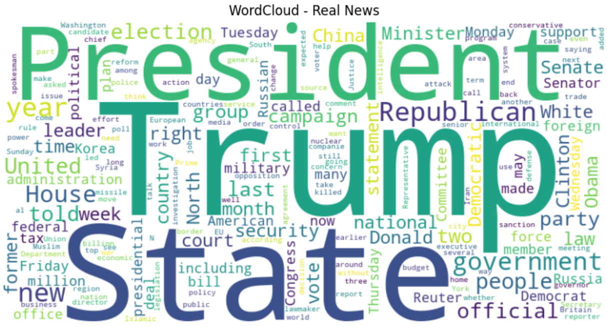

Fig. 5. Wordcloud for Real news articles.

We also performed sentiment analysis to explore whether the emotional tone of articles differs between classes. As shown in Figure 6, both fake and real news tend to have neutral sentiment on average, but fake news presents more variability.

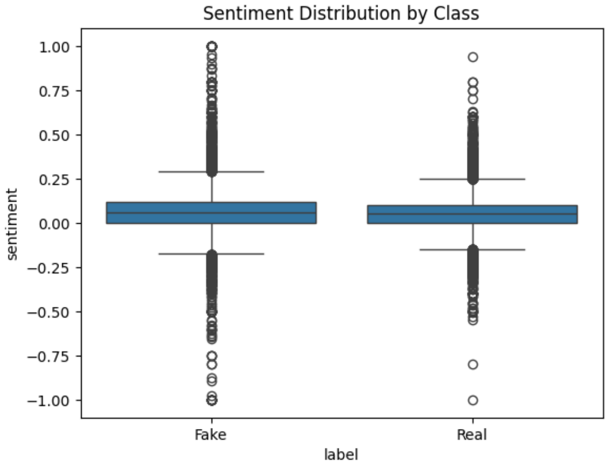

Fig. 6. Sentiment distribution for Fake and Real news.

In addition, we analyzed the distribution of topics (subject column) in both classes. Fake news covers a wide range of themes including "News," "Politics," and "Left-news," as shown in Figure 7. In contrast, real news is mostly categorized under "PoliticsNews" and "WorldNews" (Figure 8).

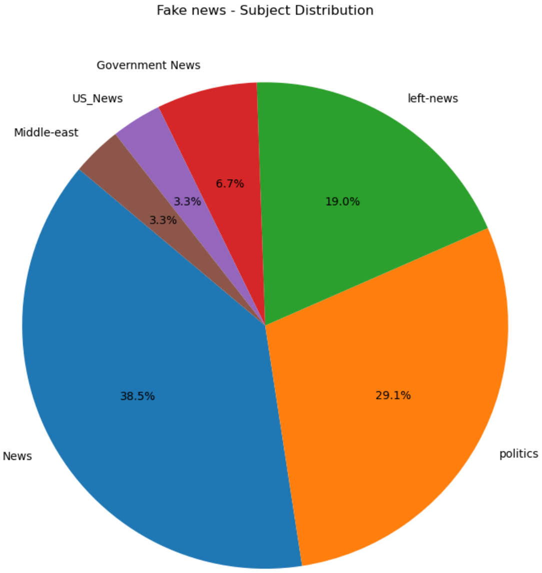

Fig. 7. Subject distribution in Fake news.

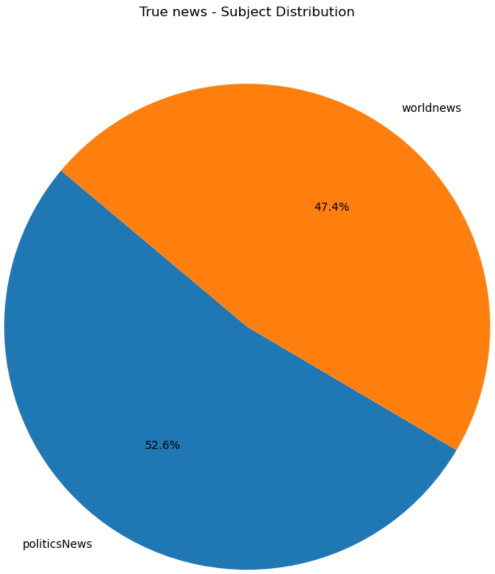

Fig. 8. Subject distribution in Real news.

Finally, we examined the number of news articles published over time. Figure 9 shows that fake news articles were more frequent earlier in the dataset, while real news became more dominant after mid-2017. This temporal trend may reflect changes in data collection or the media landscape.

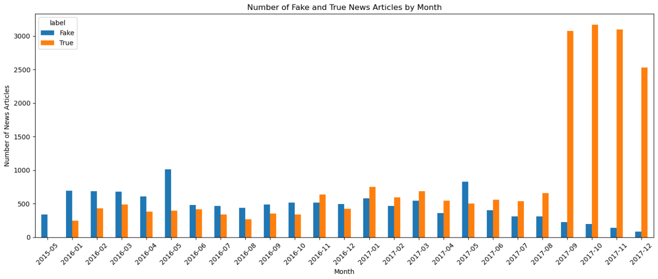

Fig. 9. Number of Fake and Real news articles per month.

This project addresses a supervised binary classification problem. The goal is to classify each news article as either fake or real. The inputs to the model are the texts of each article provided. Since these are sequences of variable length, we applied Natural Language Processing techniques to convert them into fixed-size numerical representations. The output of the model is a binary label: 0 for fake news and 1 for real news.

IV. Preprocessing and Feature Extraction

To prepare the data for training, we first removed duplicates as previously mentioned. There were 6,047 duplicate entries in the fake news dataset and 226 in the true news dataset.

Next, we cleaned the text. This involved removing source prefixes that appeared at the beginning of most articles in the true news dataset. These prefixes typically included the word "Reuters", which we used as a marker in a regular expression to identify and remove them.

After cleaning, different preprocessing steps were applied depending on the model used.

A. BERT-based Deep Learning Model

For the deep learning model, we applied preprocessing using the `BertTokenizer` from the HuggingFace Transformers library. This tokenizer breaks each article into subword units using the WordPiece algorithm, which is particularly effective for handling rare or out-of-vocabulary words.

The tokenizer converts each input text into:

- `input_ids`: numerical IDs corresponding to the subword tokens.
- `attention_mask`: a binary mask that distinguishes real tokens (1) from padding tokens (0).

To ensure compatibility with BERT's architecture, we:

- Set a maximum sequence length of 512 tokens, truncating longer texts.
- Applied automatic padding to shorter texts so all sequences have the same length.
- Left the text casing untouched, as we used `bert-base-uncased`, which automatically lowercases the input.

No additional text cleaning was performed, as the pretrained BERT tokenizer is designed to operate directly on raw text.

B. TF-IDF + Traditional Machine Learning Model

For the traditional machine learning model using TF-IDF, a simpler preprocessing approach was used:

The `TfidfVectorizer` was configured with:

- `ngram_range=(1, 2)` to capture both unigrams and bigrams.
- `max_features=20000` to limit the vocabulary size.
- `min_df=2` and `max_df=0.75` to filter out very rare and overly common terms.

While basic, this preprocessing pipeline relies on TF-IDF's built-in tokenization and normalization, which handle term frequency weighting but do not perform deep semantic understanding.

No additional preprocessing steps such as stopword removal or lemmatization were applied, as the deep learning model (based on BERT) is robust to such variations, and the TF-IDF model relies on raw word frequency patterns without semantic normalization.

For the classical models, the dataset was split into training (80%) and test (20%) subsets using stratified sampling to ensure that the class distribution (fake vs. real) remained balanced. We applied 5-fold cross-validation on the training set to assess the model's performance more robustly. After this evaluation, the final model was trained on the entire training set and tested on the held-out test set

For the deep learning model, the dataset was split into training (80%), validation (10%), and test (10%) subsets using stratified sampling. The validation set was used during training to monitor model performance across epochs, while the test set was held out and used only for final evaluation. No cross-validation was applied in this case due to the high computational cost of training large transformer-based models multiple times.

In both approaches, the output is a binary classification: label 0 for fake news and label 1 for real news. The BERT-based model produces a probability score through a sigmoid activation function, while the logistic regression model computes a probability using a linear decision boundary. In both cases, a threshold of 0.5 is used to determine the predicted class.

V. Machine Learning Models and Experimental Setup

In this project, we approached the task of fake news detection as a binary text classification problem. To better understand the effectiveness of different model families, we implemented both classical machine learning models and a transformer-based deep learning model.

A. *Classical Machine Learning Models*

We first implemented two baseline models: Logistic Regression and Support Vector Machines (SVM). Both models were trained using TF-IDF (Term Frequency-Inverse Document Frequency) vectorization to convert text into fixed-length numerical feature vectors. This approach captures word frequency while reducing the impact of very common terms.

Although these models are relatively simple, they offer useful interpretability and require minimal computational resources. Logistic Regression is effective in linearly separable cases, while SVM can capture more complex boundaries using kernel functions. However, both approaches struggle with understanding word order and deeper linguistic meaning, which limits their ability to detect more nuanced fake news.

B. *Transformer-Based Model: FakeBERT*

To overcome these limitations, we used a transformer-based model called FakeBERT, built on top of the BERT architecture. BERT (Bidirectional Encoder Representations from Transformers) is a pre-trained language model that captures context bidirectionally using self-attention mechanisms [1]. We fine-tuned FakeBERT on our dataset for binary classification by adding a classification head on top of the base model.

FakeBERT has a significantly higher capacity and can model subtle semantic and syntactic patterns. However, this comes at the cost of increased training time, larger memory usage, and the need for GPUs to achieve reasonable performance.

C. *Hyperparameter Optimization*

We performed hyperparameter optimization for both the BERT-based deep learning model and the traditional TF-IDF + Logistic Regression model in order to maximize classification performance, for the SVM we just used the default setting to compare results but it will benefit from the TF-IDF optimization.

1) BERT + CNN Model: For the BERT-based model, we evaluated combinations of the following hyperparameters:

- Dropout rate: `[0.1, 0.3]`
- Number of output channels in the convolutional layer: `[64, 128]`
- Convolutional kernel size: `[3, 5]`
- Learning rate: `[2e-5, 5e-5]`

Each configuration was trained for one epoch and evaluated using the F1-score on the validation set. The best-performing configuration was:

- `dropout=0.1`
- `out_channels=64`
- `kernel=5`
- `learning rate=2e-5`

This configuration achieved a validation F1-score of **0.9960**, indicating a very strong ability to distinguish between fake and real news.

2) TF-IDF + Logistic Regression Model: Since we had already observed that the BERT-based model yielded superior performance, we decided to explore a broader range of configurations for the TF-IDF model to determine whether it could achieve comparable results, especially given its significantly lower computational requirements.

We used a pipeline combining `TfidfVectorizer` and `LogisticRegression`, and conducted an extensive grid search over the following parameters:

- `TfidfVectorizer`:
  - `stop_words`: `['english', None]`
  - `max_df`: `[0.75, 0.85, 1.0]`
  - `min_df`: `[1, 2, 5]`
  - `ngram_range`: `[(1,1), (1,2)]`
  - `max_features`: `[5000, 10000, 20000]`
- `LogisticRegression`:
  - `C`: `[0.01, 0.1, 1, 10, 100]`
  - `solver`: `['liblinear', 'saga']`
  - `class_weight`: `[None, 'balanced']`

We used 5-fold cross-validation and selected the configuration that maximized the validation F1-score. The best configuration achieved an F1-score of **0.9904** on the validation set, with the following parameters:

- `C=100`
- `solver='liblinear'`
- `class_weight='balanced'`
- `max_df=0.75`
- `min_df=2`
- `ngram_range=(1,2)`
- `max_features=20000`
- `stop_words=None`

Despite its simplicity, the TF-IDF model showed excellent performance, nearly matching the results of the BERT-based approach, which demonstrates its effectiveness and efficiency for fake news detection when computational resources are limited.

D. ML Problem Complexity

Fake news detection is a complex problem for several reasons. First, the linguistic structure of fake and real news can be very similar, with deceptive texts often mimicking journalistic tone and style. Second, the same vocabulary can be used in both classes, making it harder for models to distinguish based solely on word frequency. Finally, fake news can evolve over time, introducing a domain shift that challenges model generalization.

While classical models provide a quick and interpretable baseline, they lack the depth needed to capture context. FakeBERT, on the other hand, handles these challenges more effectively by encoding rich contextual representations, although it requires careful fine-tuning and substantial computational resources.

VI. Results and Discussion

In this section, we present and analyze the results obtained with the models used in this project: Logistic Regression, SVM, and FakeBERT.

A. FakeBERT Results

The FakeBERT model was trained for 3 epochs using early stopping. Table I shows the metrics obtained on the validation and test sets during training.

TABLE I
FakeBERT Validation and Test Results

Epoch	Accuracy	F1 Score	Val Loss	ROC AUC
1	0.9961	0.9965	0.0113	≈ 1
2	0.9935	0.9941	0.0166	0.9999
3	0.9990	0.9991	0.0069	≈ 1
Test	**0.9992**	**0.9993**	**0.0027**	≈ 1

FakeBERT achieved a final test accuracy of 99.92% and an F1-score of 0.9993, showing excellent generalization and consistency across all evaluation metrics. The ROC AUC of almost 1, indicates strong model confidence and separation between the classes.

B. Comparison with Classical Models

To assess the benefit of using a transformer-based model, we compared FakeBERT with classical machine learning models trained on TF-IDF features.

As shown in Table II, the Logistic Regression model achieved an accuracy of 99.11%, a precision of 99.04%, a recall of 99.34%, an F1-score of 99.19%, and an ROC AUC of 99.92%. The Support Vector Machine (SVM) achieved an accuracy of 99.12%, a precision of 99.08%, a recall of 99.32%, an F1-score of 99.20%, and an ROC AUC of 99.92%. Both classical models performed remarkably well, but they still fall short of the performance achieved by FakeBERT. This confirms the ability of transformer models to better capture complex patterns in natural language, especially when detecting subtle textual cues typical of fake news.

TABLE II
Performance comparison between models

Model	Accuracy	Precision	Recall	F1 Score
Logistic Regression	0.9911	0.9904	0.9934	0.9919
SVM (Linear)	0.9912	0.9908	0.9932	0.9920
FakeBERT	0.9992	0.9991	0.9995	0.9993

The confusion matrices for the 3 models are shown in Figures 10, 11 and 12. As seen in the confusion matrices, all models perform well, especially the FakeBert model, with very few misclassifications of fake and real news.

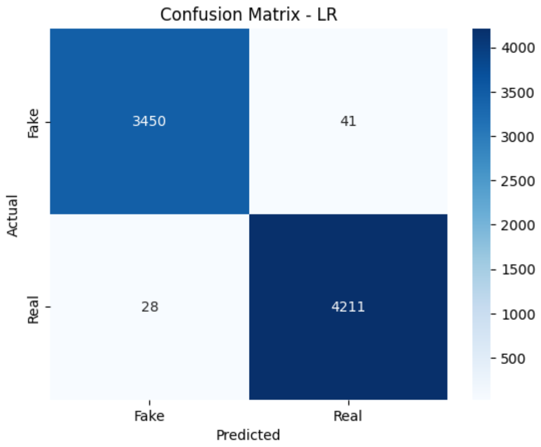

Fig. 10. Logistic Regression Confusion Matrix

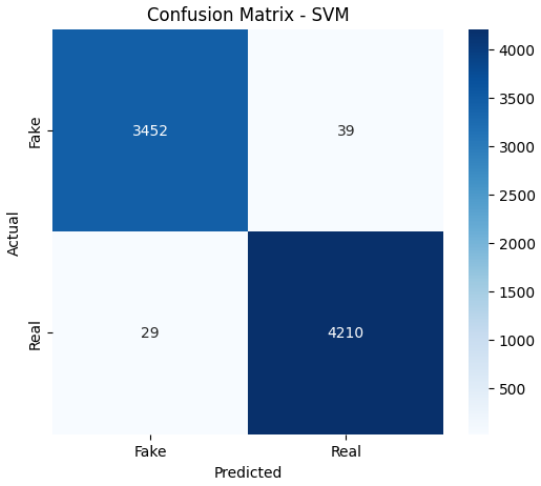

Fig. 11. SVM Confusion Matrix

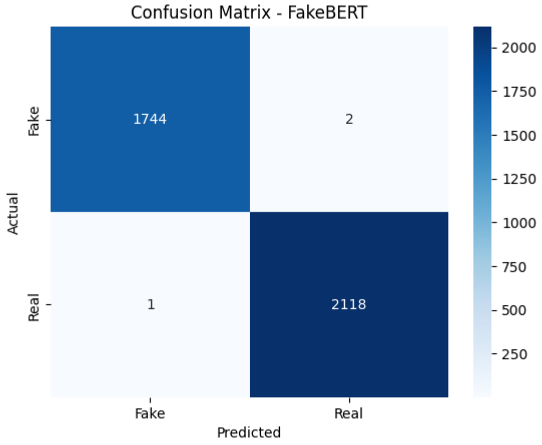

Fig. 12. FakeBERT Confusion Matrix

Although these classical models performed surprisingly well, especially the SVM, they still fall short of the performance achieved by FakeBERT. This confirms the ability of transformer models to better capture complex patterns in natural language, especially when detecting subtle textual cues typical of fake news.

C. Discussion

Fake news detection is a complex problem due to the similarity between real and fake texts, both in vocabulary and structure. Classical models like Logistic Regression and SVM are fast and interpretable but are limited to surface-level patterns. FakeBERT, on the other hand, leverages contextual embeddings and deep attention layers to capture meaning beyond simple word frequencies.

However, this performance comes at the cost of high computational demand. Training FakeBERT required GPU acceleration and significant training time (over 21 minutes per epoch). In practical applications, there is a trade-off between model complexity and performance that should be considered.

D. Comparison with Literature

To evaluate the relevance of our results, we compared them with metrics reported in recent publications on fake news detection.

In the paper “FakeBERT: Fake News Detection in Social Media with a BERT-based Deep Learning Approach” by Lav Singh et al. [5], the authors achieved an F1-score of 0.982 on a benchmark dataset using their FakeBERT model. Our implementation of FakeBERT outperforms this, reaching an F1-score of 0.9993 and an accuracy of 99.92% on the “Fake and Real News” dataset. This significant improvement can be attributed to a carefully tuned preprocessing pipeline and the integration of a CNN layer over BERT outputs, which may have helped to better capture local patterns in the embeddings.

Other studies, such as the CSI model by Ruchansky et al. [3], reported an accuracy of 89.3% on a Twitter-based dataset. While this model integrates user behavior and propagation features in addition to content, it performs considerably lower than our text-only models, especially FakeBERT. This highlights the power of transformer-based approaches in capturing textual cues, even without auxiliary information.

The classical approaches discussed in Shu et al. [4] using SVM and Logistic Regression achieved F1-scores between 0.88 and 0.93 depending on the dataset and feature representation. In our work, both classical models surpassed those figures, with F1-scores above 0.99. This may be due to differences in dataset size, preprocessing, or the careful hyperparameter tuning performed in our experiments.

In summary, our results are competitive with and in many cases superior to those found in the literature, demonstrating the effectiveness of modern transformer-based models like FakeBERT in detecting fake news.

VII. Conclusion

In this project, we addressed the problem of fake news detection using both classical machine learning models and a transformer-based deep learning model, FakeBERT. Starting from the ”Fake and Real News” dataset, we performed preprocessing, text vectorization, and model training to classify news articles as fake or real.

Our results show that FakeBERT significantly outperforms classical models such as Logistic Regression and SVM. While SVM achieved an accuracy of 99.12%, FakeBERT reached 99.92% . This improvement highlights the strength of transformer-based models in capturing subtle language patterns that simpler models may miss.

However, the enhanced performance of FakeBERT comes with higher computational requirements and longer training times. In contrast, classical models are much faster and easier to deploy, making them more suitable for applications with limited resources or real-time constraints.

For future work, it would be interesting to:

- Explore lightweight transformer variants such as DistilBERT or TinyBERT to reduce inference time.
- Test the model’s robustness using adversarial examples or intentionally manipulated fake news.
- Evaluate generalization to other datasets and domains (e.g., health-related misinformation).
- Integrate explainability tools to help understand which features contribute most to each prediction.

Overall, this project provided valuable insights into the challenges and solutions for automated fake news detection and helped us apply advanced machine learning techniques in a practical, socially relevant context.

VIII. Work Division

Artur Almeida: FakeBERT Model, LR Model, Report.
Rafael Morgado: Data analysis, SVM Model, Report.

REFERENCES

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *NAACL-HLT*, 2019.

[2] Muhammad Ali Ilyas, Abdul Rehman, Assad Abbas, Dongsun Kim, Muhammad Tahir Naseem, and Nasro Min Allah. Fake news detection on social media using ensemble methods. *Computers, Materials and Continua*, 81(3):4525–4549, 2024.

[3] Natali Ruchansky, Sungyong Seo, and Yan Liu. Csi: A hybrid deep model for fake news detection. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 797–806, 2017.

[4] Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang, and Huan Liu. Fake news detection on social media: A data mining perspective. *ACM SIGKDD Explorations Newsletter*, 19(1):22–36, 2017.

[5] Lav Singh, Neeru Sharma, Chirag Patel, et al. Fakebert: Fake news detection in social media with a bert-based deep learning approach. *Multimedia Tools and Applications*, 80(24):34507–34526, 2021.