FAKE NEWS DETECTION

Bhanu Pranaswi Sai P
S20210020309

Srinidhi M
S20210020323

Trisandhya Devi G
S20210020276

Havilah B
S20210020262

***Resumen*—ABSTRACT :**
This report explores a multimodal approach for fake news detection, integrating natural language processing and machine learning. Leveraging LSTM, SVM, KNN, Naive bayes, Random Forest, Feature Extraction and Decision Tree models, alongside visualizations like training history plots and confusion matrices, we address the challenges of identifying misinformation. Data pre-processing includes lemmatization and oversampling to enhance model robustness. Results demonstrate the effectiveness of the approach, providing insights for future research in fake news mitigation.

***Index Terms*—LSTM, SVM, KNN, Naive bayes, Random Forest, Feature Extraction and Decision Tree models.**

hyperref **CODE:**
GITHUB LINK:
Code Link
DRIVE LINK:
Code Link

I. Introduction

The proliferation of fake news has become a pervasive challenge in our information-driven society, posing a significant threat to the integrity of public discourse and the democratic process. In an era where information is disseminated at unprecedented speeds through digital platforms, the ability to distinguish between authentic and deceptive content is paramount. Our endeavor is rooted in addressing this critical concern by developing a robust machine learning model for the accurate detection of fake news.

I-0a. ***The Significance of Fake News Detection:**:* In recent years, the consequences of misinformation have reverberated across various spheres, from influencing political narratives and public opinion to sowing discord and undermining trust in media sources. The need for reliable mechanisms to combat the proliferation of fake news is not merely a technological challenge but a societal imperative. As purveyors of false information exploit the interconnected nature of digital platforms, the development of effective detection tools becomes crucial to preserving the reliability and credibility of news sources.

I-0b. ***Motivation for the Study:**:* Our motivation stems from the realization that tackling the issue of fake news requires a multidisciplinary approach, combining advancements in natural language processing, machine learning, and data analytics. The potential harm inflicted by misinformation demands a proactive stance in developing technologies that can discern subtle patterns, linguistic nuances, and contextual cues indicative of deceptive content. By undertaking this study, we aspire to contribute a practical and impactful solution to the ongoing battle against the dissemination of false information.

I-0c. ***Objectives and Methodology:**:* Our primary objective is to harness the power of supervised machine learning to create a reliable model for fake news detection. Leveraging an annotated dataset sourced from Kaggle, we employ state-of-the-art natural language processing techniques to extract meaningful features from textual content. Through an in-depth exploration of the dataset's distribution and correlation structures, we aim to uncover insights that inform our model development process.

I-0d. ***Broader Implications:**:* The implications of successful fake news detection extend far beyond academic pursuits. A proficient model has the potential to empower media organizations, social platforms, and policymakers in their efforts to combat misinformation. The model's integration into existing systems could serve as a powerful tool for enhancing the resilience of information ecosystems, fostering a more informed and discerning public.

In the subsequent sections, we delve into the specifics of our methodology, exploring data preprocessing, feature engineering, and the application of various machine learning algorithms. The journey culminates in the evaluation of our model's performance and the envisioning of a tool that can be readily deployed for real-world impact.

II. CONCEPTS USED:

Our approach encompasses a holistic methodology, from data preprocessing to model evaluation, to create a versatile and accurate Fake News Detection system. By combining the strengths of various algorithms, we aim to contribute to the ongoing efforts to combat misinformation

- **LSTM (Long Short-Term Memory):** LSTM is a type of recurrent neural network (RNN) designed to overcome the vanishing gradient problem in traditional RNNs. It is well-suited for processing and classifying sequences of data.

  II-0a. ***Application**:* LSTMs are powerful for analyzing and understanding sequences of text data, making them suitable for tasks such as natural language processing (NLP) and sentiment analysis.
- **SVM (Support Vector Machine):** SVM is a supervised machine learning algorithm used for classification and regression analysis. It works by finding the optimal hyperplane that best separates different classes in feature space.

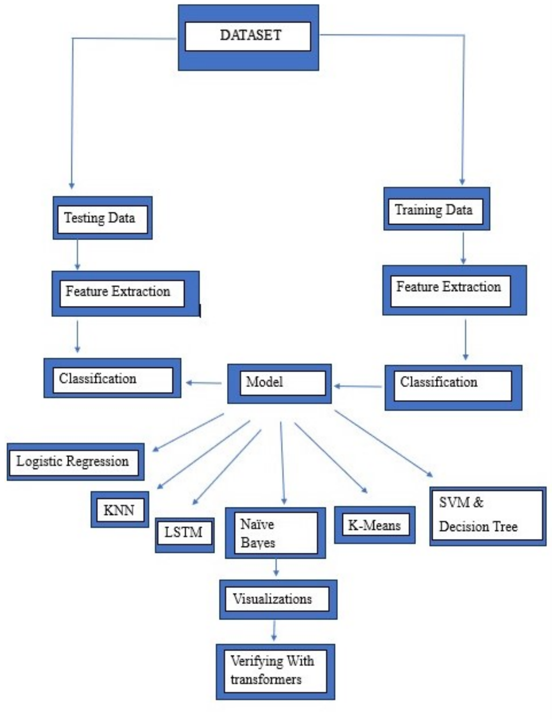


Figura 1. Block Diagram

*II-0b. **Application**:* SVM is effective for text classification tasks, as it can handle high-dimensional data and is particularly useful when dealing with non-linear relationships between features.

- **KNN (K-Nearest Neighbors):** KNN is a simple and effective supervised learning algorithm for classification and regression. It classifies a data point based on the majority class of its k-nearest neighbors in feature space.
  *II-0c. **Application**:* KNN is suitable for text classification tasks, providing a non-parametric approach to making predictions based on the similarity of instances.
- **Naive Bayes:** Naive Bayes is a probabilistic machine learning algorithm based on Bayes'theorem with an assumption of independence between features. It is simple and efficient for text classification tasks.
  *II-0d. **Application**:* Naive Bayes is commonly used for text classification, including spam filtering and sentiment analysis, due to its speed and effectiveness with high-dimensional data.
- **Random Forest:** Random Forest is an ensemble learning method that constructs a multitude of decision trees during training and outputs the mode of the classes (classification) of the individual trees.
  *II-0e. **Application**:* Random Forest is robust and effective for classification tasks. It helps to overcome overfitting and enhances the model's generalization performance.
- **Feature Extraction:** Feature extraction involves transforming raw data into a format suitable for modeling. In the context of NLP, it includes techniques like TF-IDF (Term Frequency-Inverse Document Frequency) or word embeddings to represent textual information as numerical features.
  *II-0f. **Application**:* Feature extraction is crucial for machine learning models, as it converts text data into a format that models can understand and learn from.
- **Decision Tree:** A decision tree is a tree-like model where an internal node represents a feature, the branch represents a decision rule, and each leaf node represents the outcome or class label.
  *II-0g. **Application**:* Decision trees are interpretable and useful for understanding the decision-making process. They can be employed for classification tasks and feature importance analysis.
- **Data Cleaning:** Data cleaning involves preprocessing and cleaning raw data to enhance its quality. It includes tasks such as handling missing values, removing irrelevant characters, and standardizing formats.
  *II-0h. **Application**:* Clean data is essential for building accurate and robust machine learning models. Data cleaning improves the model's ability to learn patterns and make reliable predictions.

These concepts collectively contribute to creating a comprehensive and effective Fake News Detection system, addressing challenges related to text data and providing diverse perspectives for accurate classification.

III. CONCEPT APPLICATION AND OBTAINED RESULTS

DATA ANALYSIS:

To analyze the distribution of true and fake news based on the categories and dates, we can perform exploratory data analysis (EDA) on your dataset. EDA involves summarizing and visualizing key characteristics and patterns in the data. We can analyze the distribution of news categories to understand how many articles fall into each subject and how many of them are classified as true or fake. We can examine the distribution of true and fake news over time by visualizing the count of articles for each year.

RESULTS:

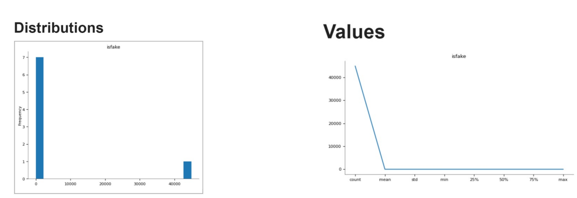


Figura 2. Distributions and Values Graphs

DATA CLEANING:

n the initial phase of the analysis, two CSV files, namely True.csv and Fake.csv, were

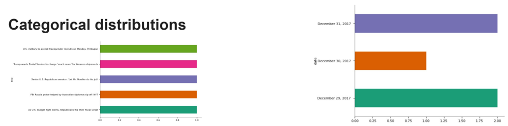


Figura 3. Categorical Distributions

loaded into Pandas DataFrames, specifically df$_trueanddf_fake.Abinarytargetcolumn,'isfake,'wasintroducedtorepresentrealnewswith0andfakenewswith1.Thedatawasconso$

For text preprocessing, common English stopwords were eliminated, and lemmatization was applied to each word using NLTK and Gensim. To address class imbalance, the minority class (fake news) was oversampled. The subsequent focus shifted to model training, where a bidirectional Long Short-Term Memory (LSTM) model was developed using TensorFlow and Keras. The text data underwent tokenization and padding before being fed into the model, which was then trained on the oversampled data with a binary cross-entropy loss function.

1. **Text Normalization:** Convert to lowercase, remove white spaces, and ensure consistent formatting.
2. **Tokenization:** Break down text into individual words or tokens for analysis.
3. **Stopword Removal:** Eliminate common, non-informative words for focused analysis.
4. **Punctuation Removal:** Remove unnecessary characters to enhance content significance.
5. **Lemmatization/Stemming:** Reduce words to their base form for normalization.
6. **Data Transformation:** Convert categorical variables into numerical format if needed.
7. **Preparation for ML:** Create a cleaned dataset suitable for machine learning model training.

RESULTS:

Original Text: This is an example sentence for text normalization. It includes various punctuations and stopwords.
Normalized Text example sentence text normalization includes various punctuation stopwords

---

FEATURE EXTRACTION:
N-grams:
N-grams are contiguous sequences of n items from a given sample of text or speech. In the context of text data, these items are usually words.
Word Embedding:
word embedding is a technique that represents words as dense vectors in a continuous vector space, capturing semantic relationships. In fake news detection, it's used to convert textual data into numerical vectors for machine learning models.

Cosine Similarity:

Cosine Similarity measures the cosine of the angle between two vectors, providing a measure of similarity. In the context of fake news detection, it's often applied to assess similarity between textual elements, like headlines and article bodies.
Readability Score:
Readability Score assesses the complexity of textual content. In fake news detection, it can offer insights into the linguistic characteristics of real and fake news, helping to distinguish between them based on writing style and complexity.

Date Conversion involves transforming date data into a standardized format, enabling chronological analysis. It aids in exploring potential temporal patterns or dependencies in fake news datasets.

These techniques collectively contribute to enhancing feature representation, analyzing temporal aspects, measuring textual similarity, and evaluating linguistic complexity, ultimately strengthening the efficiency of machine learning models in identifying fake news.

MODEL TRAINING RESULTS:

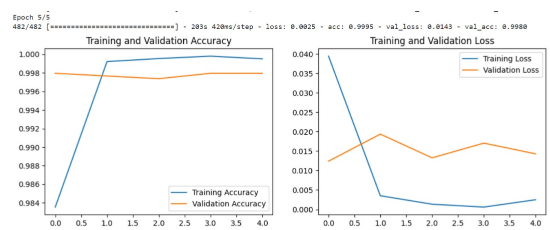


Figura 4. Training and Validation Accuracy and Loss Graphs

MODEL ACCURACY:

Model Accuracy: 0.9964981907318782

WORD CLOUD RESULTS:

SIZE OF THE WORD REPRESENTING THE FREQUENCY OF THE WORD

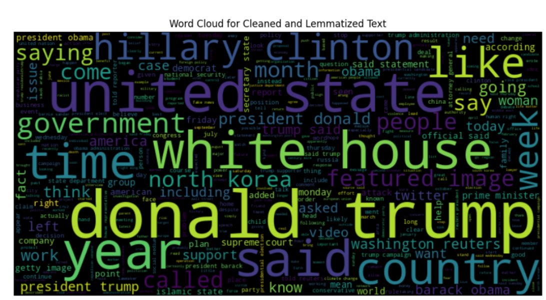


Figura 5. WORD CLOUD

CORRELATION RESULTS :

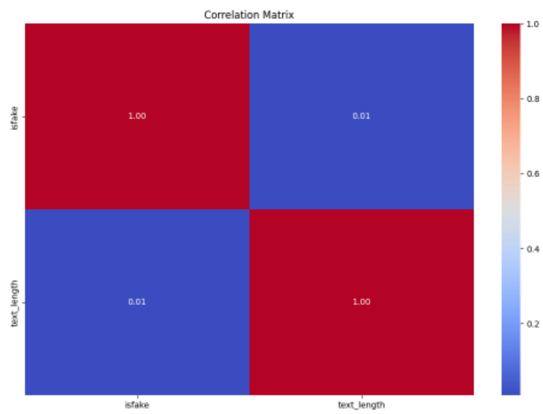


Figura 6. Correlation

In fake news detection, a correlation matrix provides crucial insights into the relationships among different features and their connection to the authenticity of news articles. The matrix reveals the degree of correlation between variables, helping us identify patterns and dependencies within the dataset. By examining feature-target correlations, we gain a clearer understanding of which characteristics are strongly associated with the likelihood of an article being fake. This information guides feature selection for machine learning models, focusing on the most relevant indicators of fake news. Additionally, correlations expose potential issues like multi-collinearity, indicating when features convey similar information. Effectively interpreting the correlation matrix aids in building more accurate and robust models, contributing to the overall effectiveness of fake news detection algorithms.

CONFUSION MATRIX RESULTS:

Following the LSTM model's training, a comprehensive evaluation was conducted, including the visualization of training history through accuracy and loss plots. The trained LSTM model was assessed on the test set, providing accuracy metrics and a confusion matrix. Subsequently, a Support Vector Machine (SVM) with a linear kernel was trained, involving TF-IDF vectorization of the text data. The SVM model's evaluation on the test set resulted in accuracy metrics and a confusion matrix.

Additionally, a Decision Tree model was trained after another TF-IDF vectorization step. The Decision Tree classifier was then evaluated on the test set, yielding accuracy metrics and a confusion matrix. Beyond these evaluations, word clouds were generated to visually represent the most frequent words in the cleaned and lemmatized text. Heatmaps were also crafted to illustrate the confusion matrices for both the SVM and Decision Tree models, offering a holistic view of the classification performance.

In our fake news detection project, we trained K-Nearest Neighbors (KNN) and Random Forest models on a labeled dataset. After preprocessing the text data, including cleaning

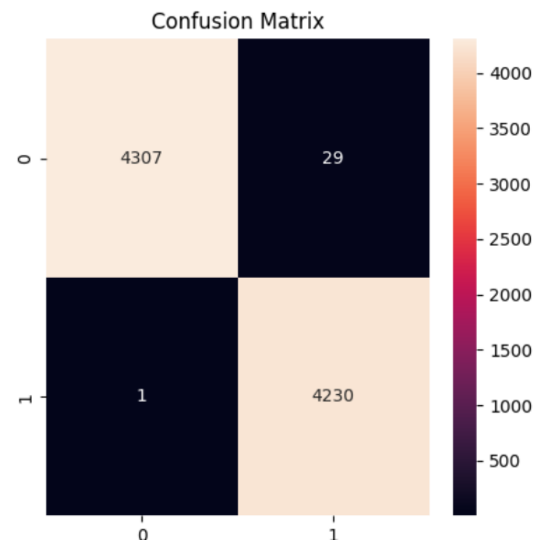


Figura 7. CONFUSION MATRIX

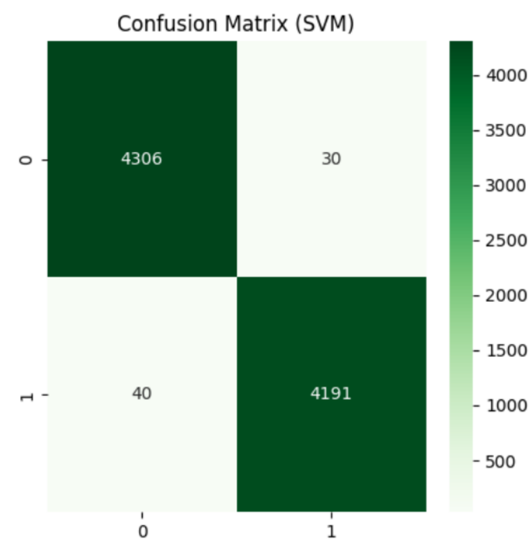


Figura 8. CONFUSION MATRIX(SVM)

and numerical conversion, we split the dataset into training and testing sets. Using the KNN algorithm (KNeighborsClassifier) and Random Forest algorithm (RandomForestClassifier), we taught the models to recognize patterns in the data. Evaluation involved assessing accuracy metrics and visualizations like confusion matrices to compare model performance. This approach helped us make an informed decision about the effectiveness of KNN and Random Forest in classifying fake news articles.

we leveraged various visualization techniques to assess model performance comprehensively. We utilized precision-recall curves, which plot precision against recall at different classification thresholds, offering insights into the trade-off between precision and recall.

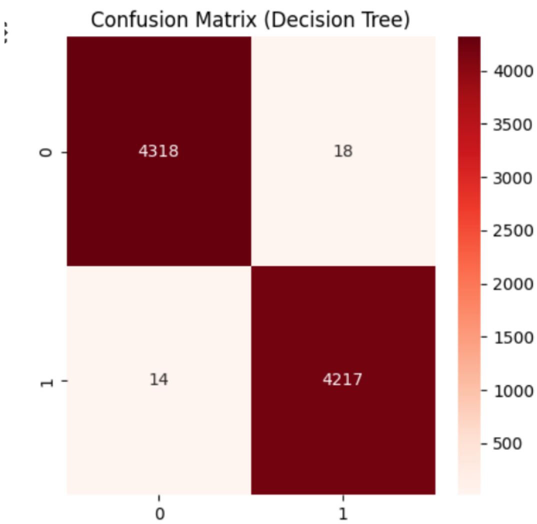


Figura 9. CONFUSION MATRIX(DECISION TREE)

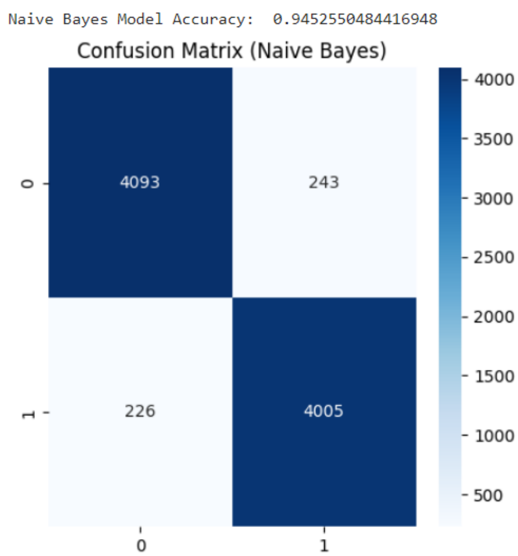


Figura 10. CONFUSION MATRIX(NAIVE BIAS)

IV. VISUALIZATION

Additionally, confusion matrices were employed to visualize true positive, true negative, false positive, and false negative predictions, aiding in a detailed evaluation of classification results. These visualizations facilitated a nuanced understanding of the strengths and weaknesses of models such as K-Nearest Neighbors (KNN) and Random Forest. Through these graphical representations, we gained valuable insights into the modelsábilities to distinguish between genuine and fake news articles.

V. LOGISTIC REGRESSION

Additionally, confusion matrices were employed to visualize true positive, true negative, false positive, and false negative predictions, aiding in a detailed evaluation of classification

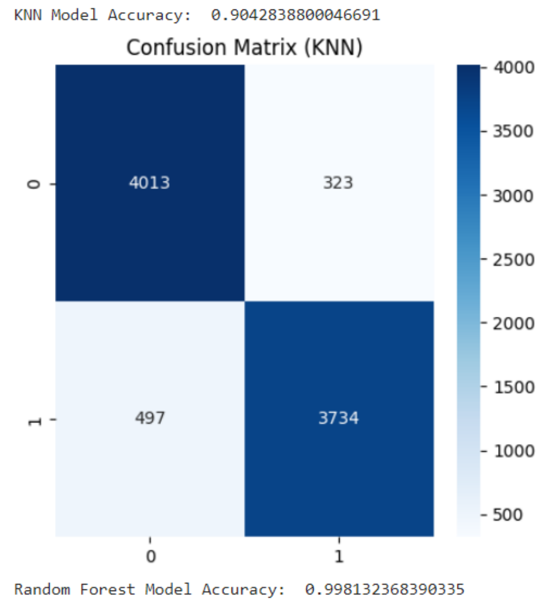


Figura 11. CONFUSION MATRIX(KNN)

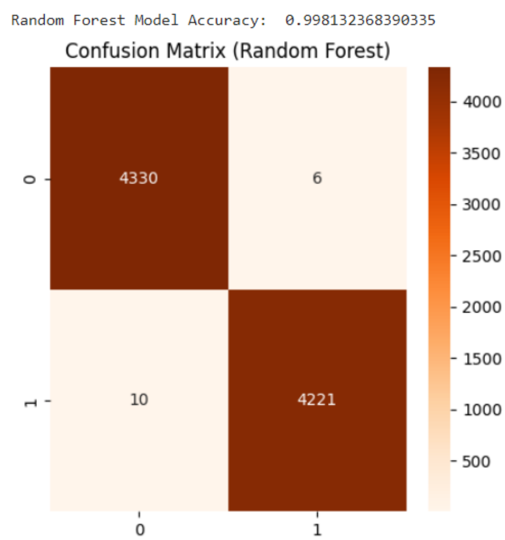
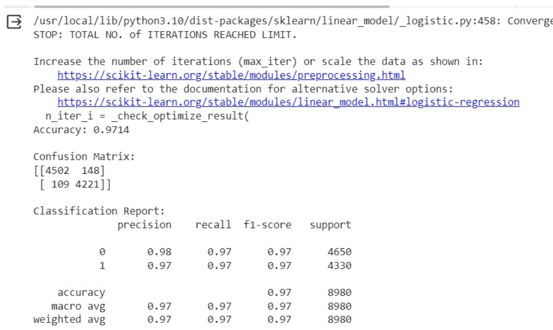


Figura 12. CONFUSION MATRIX(RANDOM FOREST)

results. These visualizations facilitated a nuanced understanding of the strengths and weaknesses of models such as K-Nearest Neighbors (KNN) and Random Forest. Through these graphical representations, we gained valuable insights into the modelsábilities to distinguish between genuine and fake news articles.

VI. TRANSFORMERS

BERT, or Bidirectional Encoder Representations from Transformers, is a groundbreaking transformer-based model designed for natural language processing (NLP) tasks. What sets BERT apart is its ability to understand the context and relationships between words bidirectionally, capturing the



```
/usr/local/lib/python3.10/dist-packages/sklearn/linear_model/_logistic.py:458: Converge
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
Accuracy: 0.9714

Confusion Matrix:
[[4502  148]
 [ 109 4221]]

Classification Report:
              precision    recall  f1-score   support

           0       0.98      0.97      0.97      4650
           1       0.97      0.97      0.97      4330

    accuracy                           0.97      8980
   macro avg       0.97      0.97      0.97      8980
weighted avg       0.97      0.97      0.97      8980
```

Figura 13. Logistic Regression output

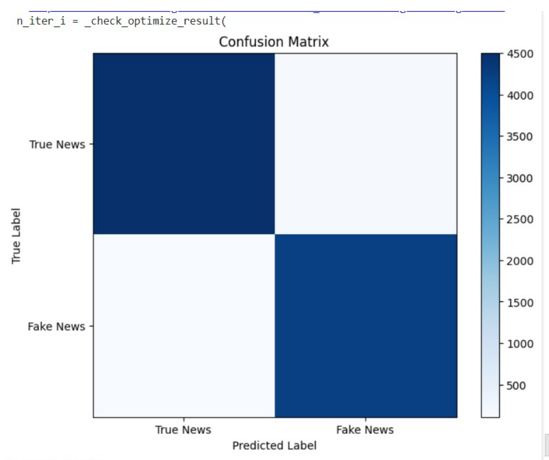


Figura 14. CONFUSION MATRIX(LOGISTIC REGRESSION)

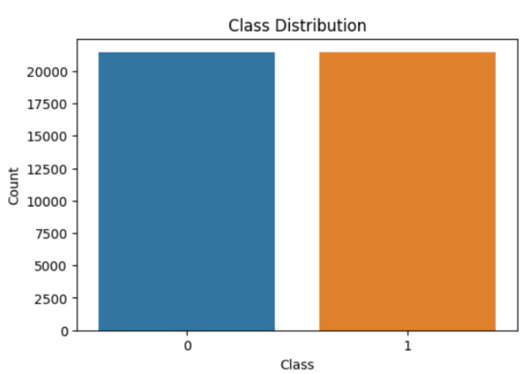


Figura 15. Graphical Visualization

intricacies of language. Pre-trained on massive corpora, BERT learns contextualized representations of words, allowing it to discern subtle nuances in meaning. In the context of fake news detection, BERT's pre-trained knowledge is leveraged through fine-tuning on specific datasets related to misinformation. This process adapts BERT to the intricacies of the fake news classification task, enabling the model to make highly accurate predictions by considering the context and semantics of the input text. BERT's success lies in its capacity for transfer learning, where the knowledge gained from a broad pre-

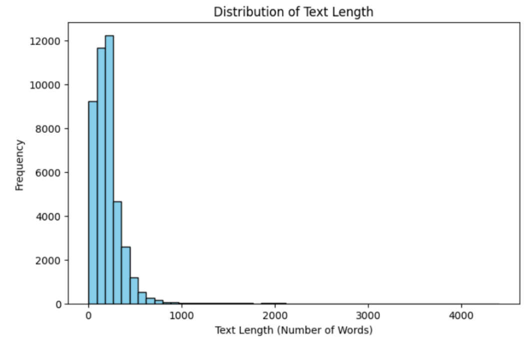


Figura 16. Graphical Visualization

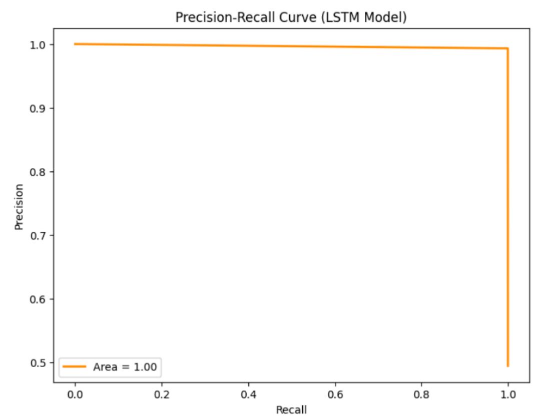


Figura 17. Graphical Visualization

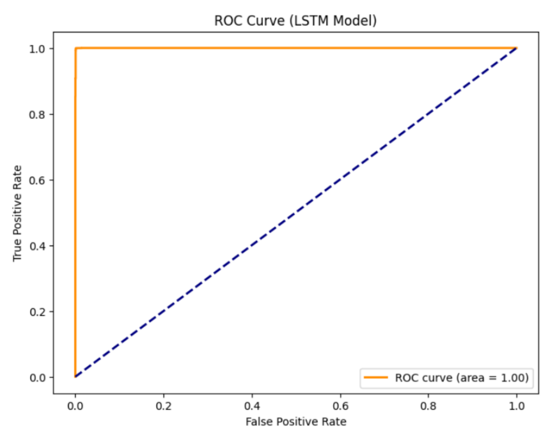


Figura 18. Graphical Visualization

training phase is transferred to and refined for the task at hand, making it a powerful tool in the realm of misinformation detection and classification.

VII. Conclusion

We explored LSTM, SVM, KNN, Naive Bayes, and Random Forest models, as well as feature extraction and decision tree techniques. The workflow involved data cleaning, normalization, and vectorization. We evaluated model performance

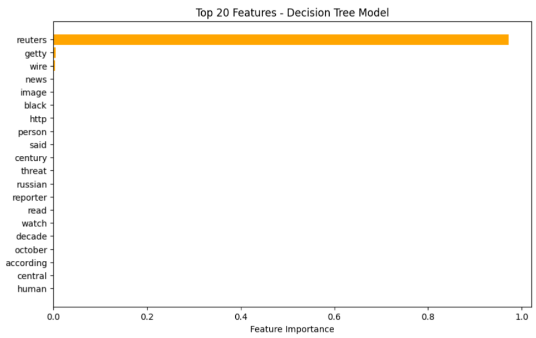


Figura 19. Graphical Visualization

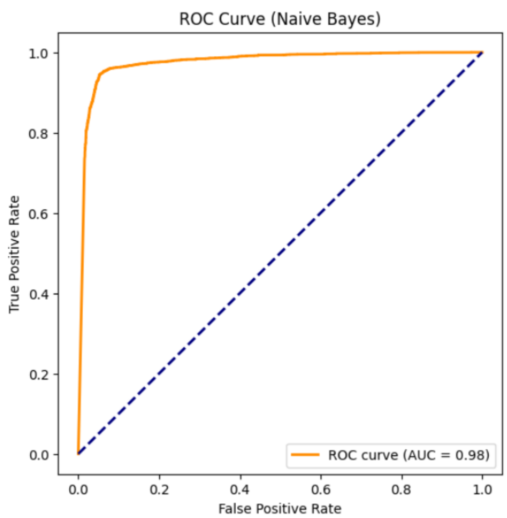


Figura 20. Graphical Visualization

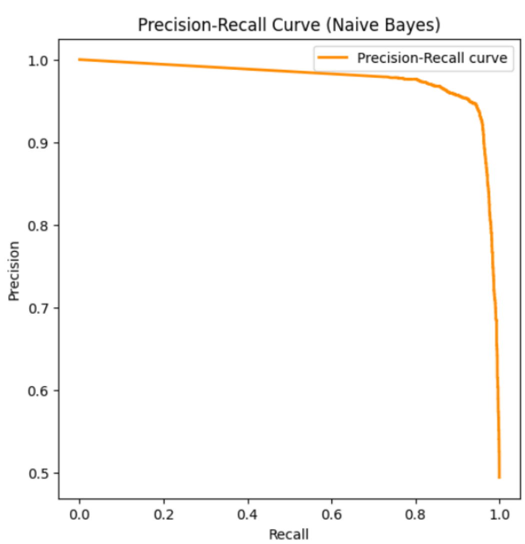


Figura 21. Graphical Visualization

using accuracy, confusion matrices, and precision-recall curves. We also touched on the application of transformers like BERT for more advanced NLP-based fake news detection.

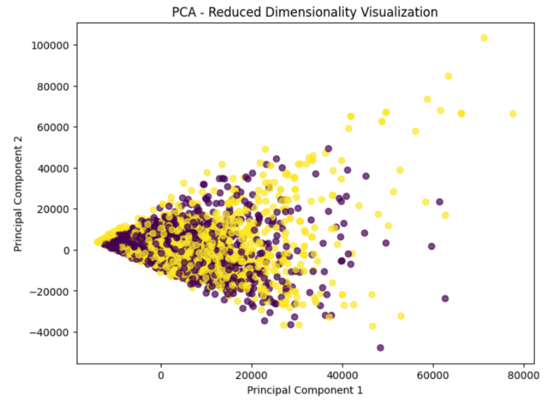


Figura 22. Graphical Visualization

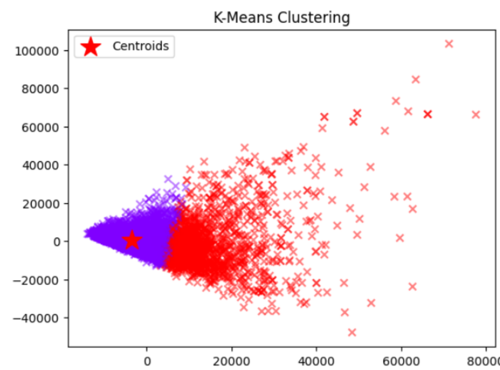


Figura 23. Graphical Visualization

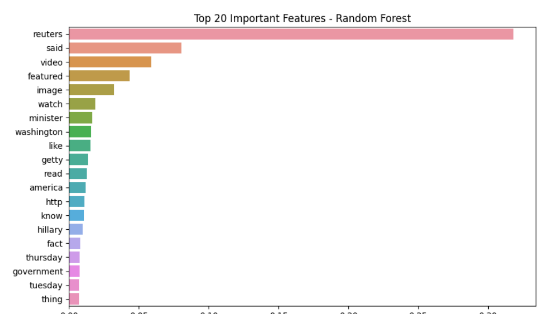


Figura 24. Graphical Visualization

Additionally, we covered data exploration, visualization, and insights into model interpretations.

```
Model Accuracy: 0.9347438752783964
Classification Report:
              precision    recall  f1-score   support

           0       0.93      0.93      0.93      4330
           1       0.93      0.94      0.94      4650

    accuracy                           0.93      8980
   macro avg       0.93      0.93      0.93      8980
weighted avg       0.93      0.93      0.93      8980

Text: Tsunami hits coastal areas, massive destruction reported!
Classification: Harmless
Percentage Harmful: 47.73%
Percentage Harmless: 52.27%

Text: Social media rumors about aliens visiting Earth
Classification: Harmful
Percentage Harmful: 89.28%
Percentage Harmless: 10.72%
```

Figura 25. Classification output

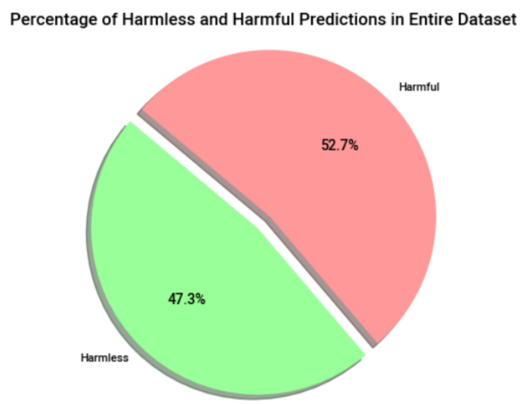


Figura 26. Graphical Visualization

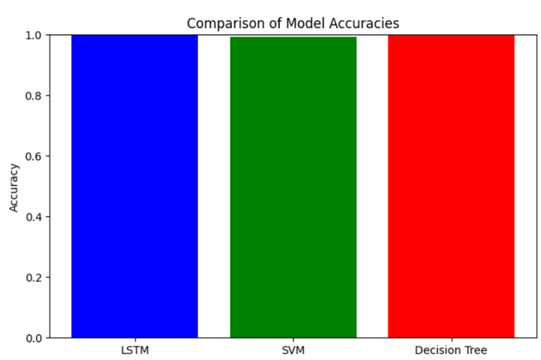


Figura 27. Comparision between 3 models

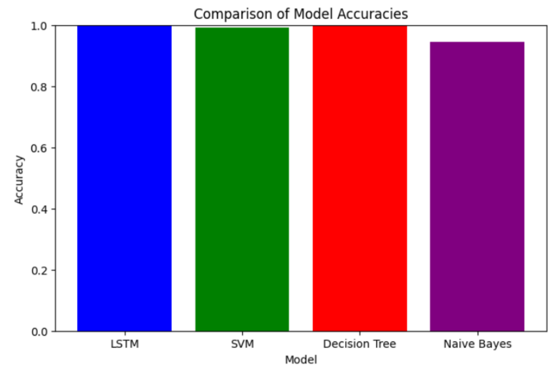


Figura 28. comparision between 4 models

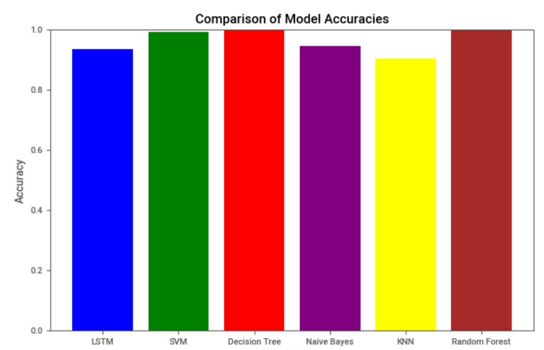


Figura 29. Comparing all the Models