News Authentication & Domain Classifier

Soham Khairnar (B23CM1039)[1] Rudra Thakar (B23EE1100)[1]
Sahilpreet Singh (B23CS1061)[1] Raman Pareek (B23EE1091)[1]
Kartik Gehlot (B23EE1088)[1] Abhishek Garg (B23EE1081)[1]

[1]IIT Jodhpur
{b23cm1039, b23ee1100, b23cs1061, b23ee1091, b23ee1088, b23ee1081}@iitj.ac.in

April 17, 2025

Abstract

In the contemporary digital age, the rapid dissemination of misinformation poses significant threats to public perception and trust. To address this challenge, we have developed a comprehensive machine learning-based application titled News Authentication and Domain Classifier. This application not only evaluates the veracity of news articles—determining whether the information is true or false—but also categorizes the news into relevant domains such as Politics, Terrorism, COVID-19, Government Affairs, among others.

Our system employs a multi-stage pipeline that enhances the reliability of predictions. Upon receiving a news article as input, the system performs real-time web search to retrieve the top five most relevant articles associated with the original input. These fetched articles are treated as novel inputs and are subjected to a series of preprocessing steps, followed by feature extraction using BERT (Bidirectional Encoder Representations from Transformers), which enables a deeper semantic understanding of the text data.

We have evaluated multiple supervised machine learning models in this project, including Support Vector Machines (SVM), Decision Trees, Artificial Neural Networks (ANN), Random Forests, XGBoost and Naive Bayes classifiers. All models were trained and tested on the IFND (Indian Fake News Dataset), chosen for its comprehensive and diverse representation of Indian news content. Among these, the Artificial Neural Network (ANN) model demonstrated superior performance in terms of accuracy and robustness.

The final classification of the authenticity is done by taking the maximum of the average probabilities for True and False for the five articles. The final prediction for category is determined through a stacking procedure employing SVM, Decision Tree & ANN. This approach not only enhances the accuracy but also mirrors human-like verification processes by consulting multiple sources before forming a conclusion.

Keywords: BERT, Decision+Tree, ANN, SVM, RBF, Stacking, Dataset, Web-Scraping, Feature+Extraction.

Contents

1 Introduction

The project addresses the problem of how the proliferation of fake news poses a serious threat to public awareness, social harmony and democratic processes. With the increasing consumption of news through online platforms, it has become crucial to develop automated systems that can verify the authenticity of information and classify its domain for better context understanding. The project has vast applications in Media & Journalism. Social Media Monitoring, Government Policy & Agencies and Academic Research.

1.1 Major Findings

1. **Web-Search** : To enable real-time information retrieval and ensure the authenticity of predictions, **SerpApi** was utilized as the primary tool for web searching. SerpApi is a real-time API service that allows access to structured search engine results from platforms like Google. In this project, it played a crucial role by fetching the top five relevant news articles based on the user's input query.

2. **Feature Extraction with BERT** : BERT was employed as a powerful tool for feature extraction to enhance the semantic understanding of news articles. Unlike traditional vectorization methods such as TF-IDF or Bag of Words, BERT captures the contextual meaning of words by analyzing them in relation to surrounding words, both left and right. This representation allows the model to better differentiate between nuanced textual patterns, which is critical when classifying news as authentic or fake and accurately identifying its domain.

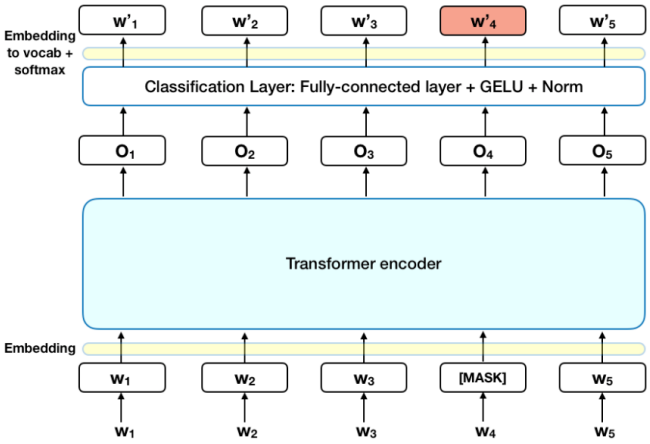

Figure 1: BERT

3. **Machine Learning Classifiers** : Various machine learning classifiers including Decision Trees, Random Forests, Support Vector Machines (SVM), Artificial Neural Networks (ANNs) & Naive Bayes were trained on the feature vectors. These classifiers exhibited varying levels of accuracy in classifying both news authenticity as well as category. All the models were able to give satisfactory results in authentication system.

4. **Authenticity System**: The authenticity verification system is designed to provide users with the flexibility to select a machine learning model of their choice for evaluating the news input. Once a model is selected, it processes the input and outputs two key results: a binary classification indicating whether the news is True or False, and a corresponding confidence score that reflects the model's certainty in its prediction.

5. **Domain Classification System**: The domain classification module was designed using a stacked ensemble approach for enhanced accuracy and robustness. Three base models—Support Vector Machine (SVM) with an RBF kernel, Articial Neural Network (ANN) , and Decision Tree—were employed, each generating a 1×9 probability vector corresponding to the nine predefined news categories. These vectors were concatenated to form a single 1×27 feature vector, which was then fed into an ANN to produce the final predicted category. This layered architecture effectively combines the strengths of multiple models, improving the system's ability to capture complex patterns in the input data.

1.2 Citing paper

Below are the references we consulted for the project :-

- [1] IFND Dataset.
- [2] Fake News Detection Using ML Approaches.
- [3] Reference Code for BERT Embedding.
- [4] Reference Code for EDA.
- [5]Scikit-learn: Machine Learning in Python

1.3 Figures

All the figures used in the report are kept in *fig* folder.

2 Approaches Tried

In this project, we experimented with various approaches to tackle the problem of image retrieval. Below are the different approaches explored along with their findings :-

2.1 Bag of Words (BoW)

- **Approach :** The Bag of Words model was used as an initial method for text vectorization. Each news article was transformed into a sparse vector representing the frequency of words occurring in the text, without considering the order or semantics.
- **Findings :** BoW worked decently on relatively short and syntactically simple news inputs, allowing models to distinguish between basic patterns. Simpler classifiers, especially Naive Bayes, showed better performance with BoW compared to contextual embeddings.
- **Problems Faced :** The major limitation of BoW was its inability to capture the context and semantics of the text, which is crucial in news analysis. The high dimensionality of the resulting sparse vectors also posed challenges like increased memory usage and overfitting. Moreover, performance degraded for longer and contextually complex inputs, leading to a shift toward BERT embeddings in the final system.

2.2 K-Nearest Neighbors (KNN)

- **Approach :** Implemented for both authenticity detection and domain classification. After preprocessing and feature extraction using BERT embeddings, the vectors were passed to the classifier which classifies each input based on the majority label among its 'K' closest neighbors in the feature space, with Euclidean distance used as the distance metric.
- **Findings :** The KNN model demonstrated moderate performance, particularly in simpler classification scenarios. It was able to provide reasonably accurate results when the feature space was not highly complex or overlapping.
- **Problems Faced :** The model was inefficient with high-dimensional BERT feature vectors, leading to increased computational time and reduced scalability. It also struggled with overlapping categories in domain classification due to its lack of a learning mechanism. As a result, it was scraped off.

2.3 Random Forest (RF)

- **Approach :** Random Forest is an ensemble learning method that constructs multiple decision trees during training and outputs the mode of the classes (classification) or mean prediction (regression) of the individual trees. allowing the model to generalize better and reduce overfitting.

- **Findings :** Random Forest performed consistently well and was particularly effective in the domain classification module, producing stable probability distributions across all categories. It had the ability to handle high-dimensional data to avoid overfitting. We had also tried Random Forest with *PCA* which gave satisfactory results.

- **Problems Faced :** Although Random Forests improved generalization, they were computationally intensive and slower in training due to the large number of decision trees. Additionally, interpreting the results was more complex compared to single decision trees, making it harder to analyze misclassifications.

2.4 XGBoost

- **Approach :** XGBoost, an ensemble learning method based on gradient boosting, was implemented to evaluate its ability to classify domain categories. The feature vectors generated from BERT were used as input, and hyperparameters like learning rate, max depth, and number of estimators were tuned using grid search to optimize performance.

- **Findings :** XGBoost performed competitively in the classification, showing robustness in handling noisy data. It delivered accuracy close to Random Forest but lagged slightly behind ANN and SVM for multi-class domain classification.

- **Problems Faced :** The model was sensitive to hyperparameter tuning and prone to overfitting in certain configurations, particularly on imbalanced category data. Additionally, its interpretability was limited due to the complexity of the ensemble structure, making it less transparent compared to decision trees.

2.5 Naive Bayes (NB)

- **Approach :** A simple probabilistic model tried for both authentication and Domain classifier systems. It was implemented using Bayes theorem.

- **Findings :** Naive Bayes achieved relatively lower performance compared to other models. It struggled with the dense and context-rich BERT embeddings, which violated the independence assumptions. However, it remained fast and computationally efficient, making it suitable for quick prototyping and small-scale experiments.

- **Problems Faced :** The core limitation was the model's assumption of feature independence, which does not hold true for semantic embeddings generated by BERT. This led to poor generalization and reduced accuracy in both classification tasks. As a result, the model was scraped off in this final application.

2.6 Decision Tree (DT)

- **Approach :** A single Decision Tree classifier was trained using the extracted BERT features. The model learned simple if-else rules by recursively splitting the dataset based on Gini index to perform the requested tasks.

- **Findings :** Decision Tree models were able to capture non-linear decision boundaries to a certain extent and provided interpretable decision logic. However, their standalone performance was inferior to ensemble methods like Random Forests and neural networks.

- **Problems Faced :** The model exhibited overfitting on the training data, particularly in high-dimensional feature spaces. Its performance on unseen data was inconsistent, and minor variations in the data led to significant changes in the structure of the tree, affecting robustness.

2.7 Support Vector Machines (SVM) with RBF Kernel

- **Approach :** Implementation of SVM with Radial Basis Function (RBF) kernel allowed the model to map the non-linearly separable BERT-based high-dimensional vectors into a higher-dimensional feature space for effective classification. The model was tuned for hyperparameters such as the regularization parameter C and kernel coefficient γ.

- **Findings :** SVM showed solid performance on relatively clean and well-separated classes. It produced reliable predictions for domain classification and was especially useful in our stacking setup. It was also computationally more efficient than ANN during training for moderate-sized datasets.
- **Problems Faced :** The main challenge was scaling SVM for large datasets, as training time and memory usage increased significantly. It also lacked probabilistic outputs natively, which required calibration to generate confidence scores, adding complexity to the pipeline.

2.8 Artificial Neural Network (ANN)

- **Approach :** The custom feedforward ANN consisted of multiple dense layers with ReLU activation, dropout for regularization, and a softmax output layer. It received either BERT-extracted feature vectors or the stacked ensemble vectors as input.
- **Findings :** ANN outperformed all other individual models in both tasks, demonstrating high accuracy and confidence scores. It was especially effective in the final ensemble for domain classification, learning complex patterns from the stacked model outputs.
- **Problems Faced :** ANN required substantial training time and was prone to overfitting without proper regularization. Fine-tuning the architecture and hyperparameters (like learning rate, number of layers, and neurons) was a trial-and-error process that required multiple iterations.

2.9 Stacking

- **Approach :** Stacking was used to improve the performance of domain classification. Three models—SVM with RBF kernel, ANN, and Decision Tree—were used as base learners. Each generated a probability distribution across 9 classes, producing a concatenated 1×27 vector. This vector was then passed to another ANN, which acted as a meta-learner to make the final classification.
- **Findings :** The stacked architecture significantly improved the robustness and generalization of the domain classifier. It leveraged the diversity among the base models, and the ANN effectively captured their combined strengths, yielding high accuracy and confidence.
- **Problems Faced :** The increased complexity led to higher computation time and required careful synchronization between base learners. Overfitting was a potential issue if the base models were too correlated, and feature scaling consistency had to be maintained across all models.

3 Experiments and Results

In this section, we detail the experiments conducted to develop and evaluate the News Authentication & Domain Classifier System. The experiments involved various stages, including data preprocessing, feature extraction, classifier training, and the final working of the project.

3.1 Experimental Setup

The Experimental Setup comprised the following steps :-

1. **Dataset :** The IFND dataset covers news pertaining to India only. This dataset is created by scraping Indian fact-checking websites. The dataset contains two types of news : Fake and Real news. The dataset contains different types of articles on different topics; however, the majority of news focuses on political news. The other news categories are: Covid-19, Violence, Terror, Government, Trade & Misleading. The dataset contains 56715 rows and 6 columns: ID, Statement, Image, Date of News, Category, Label (True / False). We analyzed the dataset using various statistical and visual techniques to gain insights into its structure and characteristics. EDA - Exploratory Data Analysis - helps in understanding the distribution of data, detecting patterns, identifying anomalies, and preparing data for modeling.
2. **Data Preprocessing :** We obtained the IFND Dataset and pre-processed the text by changing the case to lowercase , removing numbers , punctuation marks and special characters.

3. **Feature Extraction :** We used BERT for feature extraction from the pre-processed text. BERT converted the test to a feature vector by capturing the contextual meaning of words by analyzing them in relation to surrounding words, both left and right.

4. **Dimensionality Reduction :** To reduce the dimensionality of the extracted features, PCA was employed with to transform the feature vectors into lower-dimensional spaces while preserving relevant information. It was employed to be used with Random Forest.

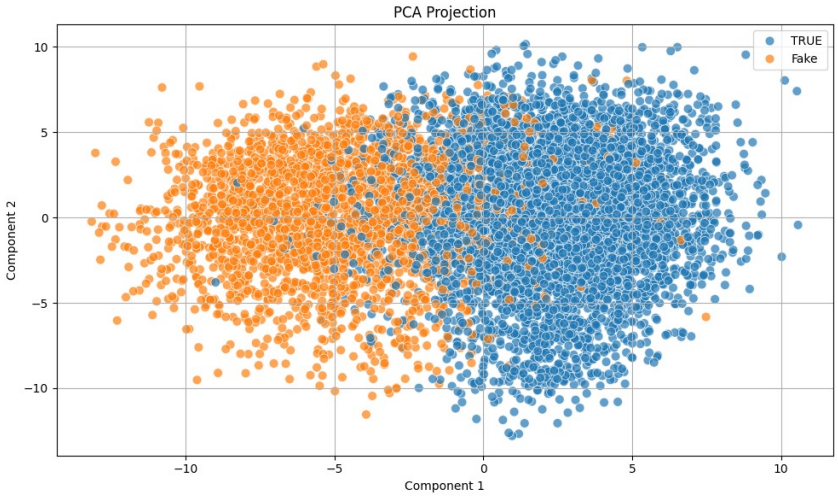

Figure 2: PCA

3.2 Analysis

A comprehensive evaluation was done for the models implemented, their comparative performance, and key observations were noted. The following table shows the performance metrics (weighted average) of the models tried :-

Table 1: Performance Comparison of Different Models on True / False

Model	Accuracy	F1 Score	Precision	Recall
K-Nearest Neighbors (KNN)	73.3%	0.82	0.72	0.97
Random Forest + PCA	94.47%	0.94	0.93	0.97
Naive Bayes	89.37%	0.89	0.90	0.89
Decision Tree	89%	0.88	0.87	0.89
SVM (RBF Kernel)	94%	0.94	0.94	0.94
Artificial Neural Network (ANN)	**94.42%**	**0.94**	**0.92**	**0.95**

Table 2: Performance Comparison of Different Models on Category

Model	Accuracy	F1 Score	Precision	Recall
Random Forest	52.07%	0.51	0.58	0.52
Naive Bayes	65.5%	0.65	0.66	0.66
Decision Tree	59.11%	0.59	0.59	0.59
SVM (RBF Kernel)	71%	0.70	0.71	0.70
Artificial Neural Network (ANN)	74.7%	0.75	0.73	0.76
Stacked Model	**83.6%**	**0.84**	**0.85**	**0.82**

From the table, we can see that Artificial Neural Network (ANN) excels in predicting the authenticity of the news. The Stacked model due to its generalized nature and robustness, gave better results in predicting the domain of the news. Due to relatively lower performance, K-Nearest Neighbors (KNN) was excluded from the final integration phase involving web-based predictions.

3.3 Methodology

Depending on the results given by all the models, specific models were chosen and were integrated with the web-based predictions. The entire pipeline is shown in three different steps :-

1. **Data Processing :** This step involved taking input from the user, followed by web-searching and extracting top 5 news articles using SerpApi. The retrieved news articles and processed using BERT to get the final 5 feature vectors. The user also has the liberty to choose any of the four models for classification - ANN , SVM (RBF) , Decision Tree & RF.

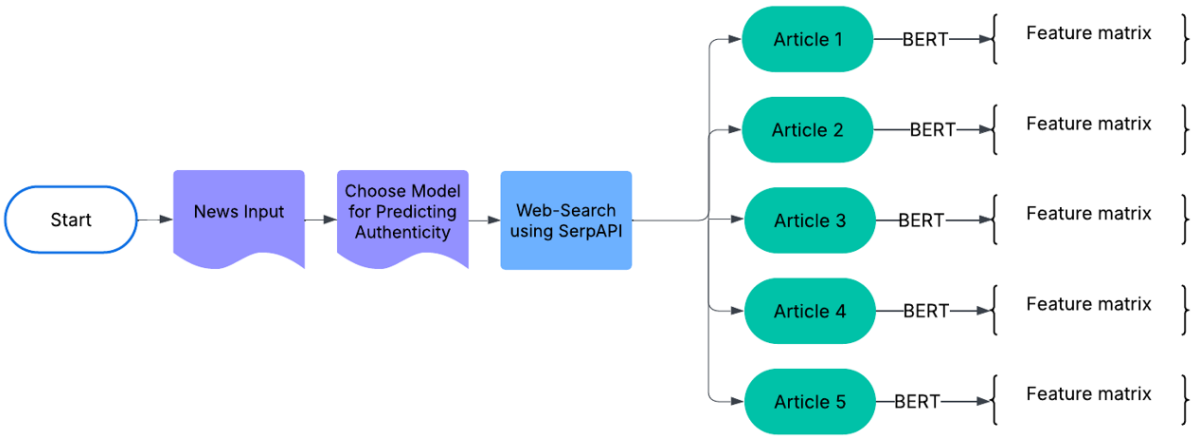

Figure 3: Data Processing

2. **Authenticity Prediction :** The processed feature matrix is fed into the model selected by the user. For each of the five retrieved articles, the model generates a probability matrix indicating the likelihood of the article being true or false. These five probability matrices are then aggregated by computing the average probabilities for both classes. The class (True/False) with the higher average probability is presented as the final prediction. Additionally, a confidence score—defined as the average probability corresponding to the predicted label—is displayed alongside the result.

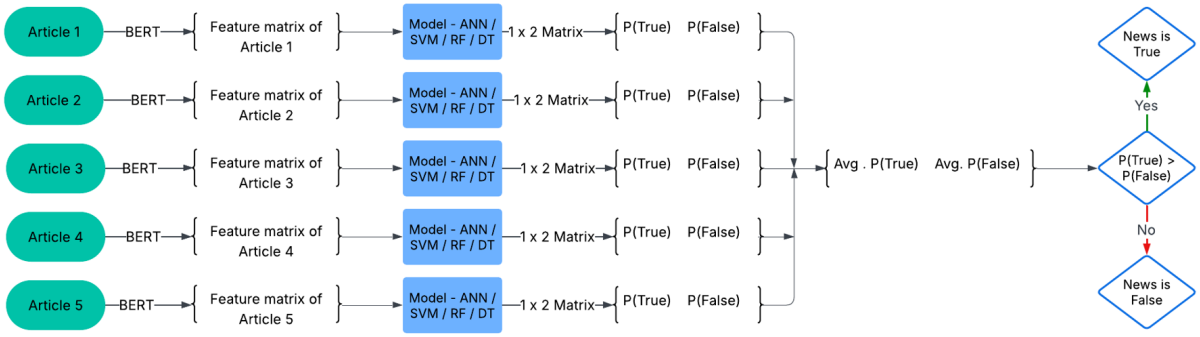

Figure 4: Authenticity Prediction

3. **Domain Prediction using Stacking :** The processed feature matrix is passed through three base models—Artificial Neural Network (ANN), Support Vector Machine (SVM) with RBF Kernel, and Random Forest (RF). Each model outputs a 1×9 probability vector, representing the likelihood of the input belonging to each of the nine predefined categories. These vectors are concatenated to form a single 1×27 feature vector, which is then used as input to a final ANN classifier. This meta-model outputs the predicted category along with a corresponding confidence score.

4. **Use of Google Cloud :** Google Cloud services were explored and utilized for application deployment. Its scalable infrastructure provided a reliable environment for hosting components of the system and experimenting with cloud-based data retrieval and processing workflows.

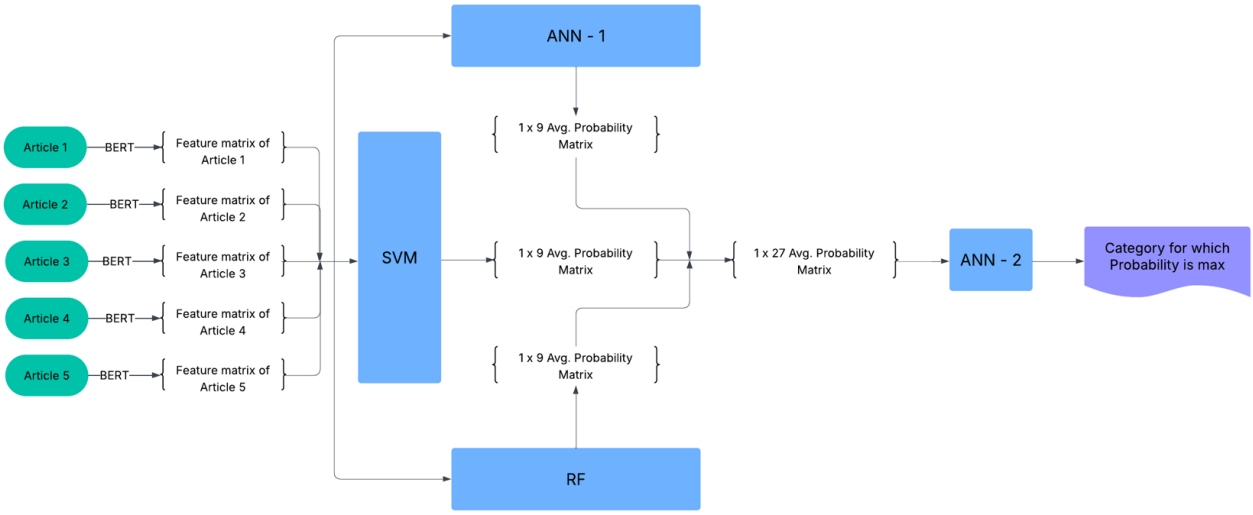

Figure 5: Domain Prediction using Stacking

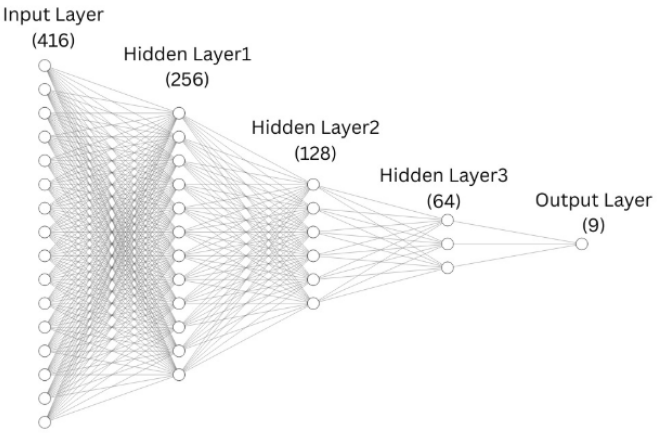

Figure 6: Architecture of ANN - 1

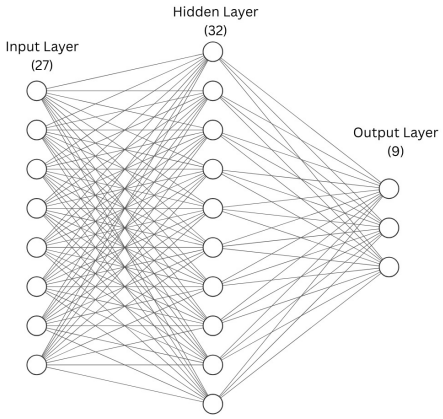

Figure 7: Architecture of ANN - 2

3.4 Results

The final application was deployed using **Google Cloud** and tested using a wide range of news inputs across various domains such as politics, COVID-19, government affairs, and terrorism. The system successfully performed both authenticity verification and domain classification with high reliability. For each input news statement, the application retrieved relevant articles via web search, processed them through the feature extraction pipeline, and delivered accurate predictions along with confidence scores. The results were visualized on a user-friendly web interface, clearly displaying the predicted label (True/False), the most probable category, and the confidence associated with each. The screenshots of the same are shown below :-

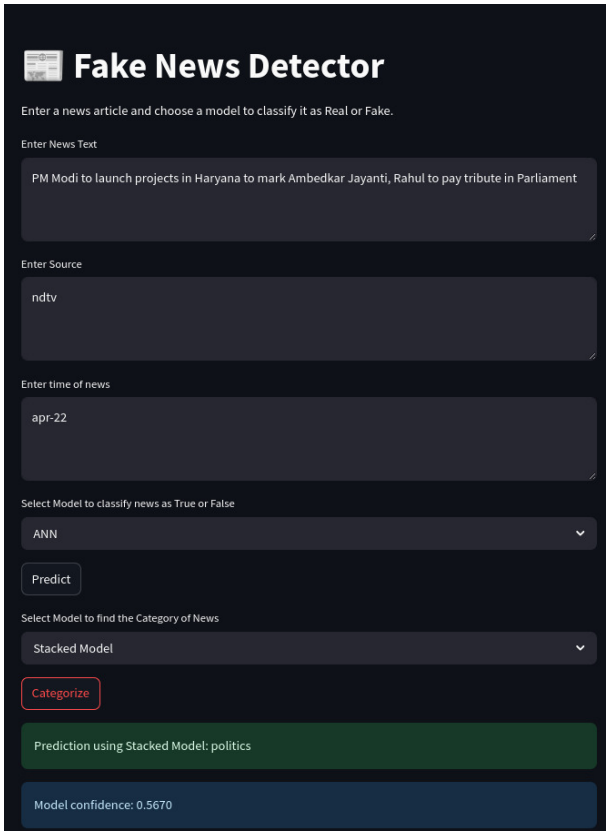

Figure 8: Authenticity Prediction

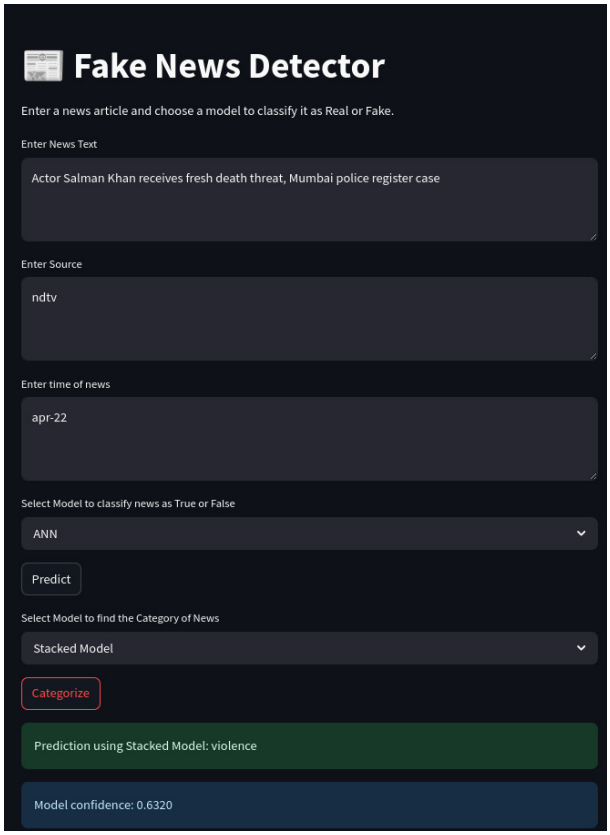

Figure 9: Domain Prediction

3.5 Limitations

Despite its promising performance, the project has a few notable limitations.

- The system's reliance on web search through SerpApi means its accuracy is inherently tied to the quality and relevance of retrieved articles, which may vary based on search engine fluctuations or limited data availability for niche topics.
- The models are trained on the IFND dataset, which, while comprehensive, may not fully represent the linguistic and contextual diversity present in global news. This could lead to generalization issues when exposed to real-world or multilingual inputs.
- Domain classification is restricted to a fixed set of predefined categories, limiting flexibility in capturing emerging or nuanced news topics.
- The system also assumes textual data; hence, it does not currently analyze multimedia content, which is increasingly prevalent in digital news.
- The performance of stacked models increases computational complexity, which could hinder scalability in low-resource environments or real-time applications.

3.6 Future Scope

The future scope of the application includes several promising directions. The system can be extended to support real-time verification of live news feeds or social media content, enabling quicker identification of misinformation. Incorporating multilingual capabilities through models like mBERT would allow for broader applicability across global news sources. Future work may also explore integrating multimedia analysis to detect manipulated images or videos accompanying the news. Additionally, a feedback-driven loop could be introduced to improve model accuracy over time, while mobile or browser-based versions can enhance accessibility and user engagement.

4 Summary

- The IFND dataset was acquired and preprocessed, after which contextual feature extraction was performed using BERT. Additionally, Principal Component Analysis (PCA) was applied in the Random Forest pipeline to reduce dimensionality and enhance computational efficiency.
- Multiple machine learning classifiers, including Artificial Neural Networks (ANN), Support Vector Machines (SVM) with RBF kernel, Random Forest, Decision Trees, and Naive Bayes, were trained, validated, and compared using metrics such as accuracy, precision, recall, and F1 score.
- A web-based pipeline was incorporated using SerpApi to dynamically retrieve relevant articles for any given news input. The model evaluated these articles and provided predictions based on aggregated probability scores.
- For domain classification, a stacked architecture was designed using ANN, SVM, and RF as base models, whose outputs were fused and passed to a final ANN classifier, significantly improving performance.
- The final system enables users to choose their desired model for authenticity checks and displays outputs in a clear interface with confidence scores and predicted categories.
- Based on experimental analysis, ANN emerged as the most effective model in terms of overall performance, showing high accuracy and consistency across both classification tasks.
- The project demonstrates the practical use of ensemble techniques, web integration, and deep NLP-based feature extraction in building real-world misinformation detection tools.

References

[1] Sonal Garg. IFND Dataset. https://www.kaggle.com/datasets/sonalgarg174/ifnd-dataset, 2023. Accessed: 2025-04-14.

[2] Z Khanam, B N Alwasel, H Sirafi, and M Rashid. Fake news detection using machine learning approaches. *IOP Conference Series: Materials Science and Engineering*, 1099(1):012040, mar 2021. doi:10.1088/1757-899X/1099/1/012040. URL https://dx.doi.org/10.1088/1757-899X/1099/1/012040.

[3] Anand Mishra. Reference code. https://github.com/anandmishra22/PRML-Spring-2023/blob/main/Project/Reference_Codes/refCodes4PRMLProject.ipynb, 2023.

[4] Soha Mohajeri. BuzzFeed News Analysis and Classification. https://www.kaggle.com/code/sohamohajeri/buzzfeed-news-analysis-and-classification/notebook, 2021.

[5] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Andreas Müller, Joel Nothman, Gilles Louppe, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine learning in python, 2018. URL https://arxiv.org/abs/1201.0490.

A Contribution of each member

1. Rudra Thakar : Implemented the stacking architecture & ANN.
2. Sahilpreet Singh : Website Deployment and performed feature extraction & data preprocessing.
3. Soham Khairnar : Implemented SVM and performed EDA on the dataset.
4. Kartik Gehlot : Implemented Decision Tree and prepared the video
5. Raman Pareek : Implemented Random Forest and dimensionality reduction on extracted features - PCA.

6. Abhishek Garg : Implemented Naive Bayes and XGBoost.

Collaborative Effort : Report, ReadMe, Minutes of Meetings, Project Page and Webpage was done by the collective effort of all.