Harnessing Machine Learning for Detecting False Information

Komal Arora
Assistant Professor
Lovely Professional University, Punjab
arorakomal446@gmail.com

Aparajit Sharan
Dept. of Computer Science & Engineering
Lovely Professional University, Punjab
sharanaparajit19@gmail.com

Tushar Bharti
Dept. of Computer Science & Engineering
Lovely Professional University, Punjab
tushar.rooks@gmail.com

Rishabh Shukla
Dept. of Computer Science & Engineering
Lovely Professional University, Punjab
rishabh×shukla@gmail.com

Himanshu Vyas
Dept. of Computer Science & Engineering
Lovely Professional University, Punjab
mandyhiman918 @gmail.com

Prathamesh Mistry
Dept. of Computer Science & Engineering
Lovely Professional University, Punjab
prothomeshmistry@gmail.com

ABSTRACT—

In today's digital era, people heavily rely on a plethora of online sources for staying updated with current events, owing to the ubiquitous presence of the internet. The extensive utilization of social media's, including Twitter, Facebook, and other has facilitated the swift circulation of information among vast numbers of individuals in a brief timeframe. The ramifications of spreading misinformation are significant, spanning from impacting election results to moulding prejudiced viewpoints. Various platforms such as WhatsApp, Instagram, and many others play pivotal roles as conduits for the dissemination of false information. To address this concern, the present research suggests a remedy through the implementation of a machine learning-driven system for detecting fake news. This approach entails initially extracting data from diverse news sources utilizing web scraping methodologies, subsequently organizing the acquired data into separate datasets, specifically classified as authentic and fabricated news items. The classification of data employs a variety of algorithms including Naïve Bayes, K-Nearest Neighbour, Logistic Regression (LR), Random Forest, Support Vector Machine (SVM) and Decision Tree. Based on model's output, articles of news are classified as either genuine or fabricated. Users can access this functionality through a web server to verify the authenticity of news content.

I. INTRODUCTION

The term 'Misinformation' denotes the dissemination of false or misleading information, often through unverified sources or fabricated content. Throughout history, misinformation has persisted in various forms, such as gossip and rumours. Social media platforms serve as conduits for the rapid spread of misinformation, facilitated not only by human users but also by automated bots. These bots play a significant role in amplifying the reach and impact of false information. Efforts to combat misinformation aim to prevent its proliferation across different platforms, including social media and messaging services. The motivation behind such endeavours lies in preventing the potentially harmful consequences of misinformation, which can incite violence and lead to tragic incidents like lynchings and riots. Detection systems are crucial in identifying and thwarting the spread of false information, thereby safeguarding society from the negative repercussions of such acts. One approach to addressing this issue involves developing systems capable of verifying the authenticity of news articles. These systems utilize data gathered through web scraping techniques to classify news items as either true or false. Different types of classification models, such as Random Forest, Decision Trees, K-Nearest Neighbours (K-NN) etc., are utilized to bolster predictive accuracy. Furthermore, an investigation into amalgamating

these models is conducted to enhance overall performance.

The format of this document is organized as follows: In Section 2, a comprehensive review of existing studies pertaining to misinformation detection is presented. Following this, Section 3 provides an outline of the process of data extraction, pre-processing, and the selection of classifiers. Section 4 presents the accuracies of the classifiers and their associated findings. Finally, Section 5 provides concluding remarks on the study's outcomes.

II. PRIOR STUDIES

Several research initiatives have aimed to tackle the pervasive issue of fake news detection, employing various methodologies and algorithms. Granik et al. introduced a straightforward approach utilizing a naive Bayes classifier, achieving a commendable accuracy of approximately 74% when tested on a dataset comprising Facebook posts from both mainstream political news pages and other sources. However, challenges arose due to the dataset's imbalance, with a minority of posts categorized as fake news.

In a distinct research endeavour, Shakya and colleagues introduced a comprehensive approach to identifying fabricated news, integrating aggregator utilities to gather news articles from various origins for efficient examination. Their approach involved utilizing a combination of classification techniques, such as Naive Bayes and SVM, augmented with semantic analysis to address the multifaceted nature of fake news. Furthermore, researchers have explored the intricacies of data cleaning techniques, emphasizing the importance of error detection and repair to enhance data quality. Leveraging machine learning approaches, they underscored the significance of preprocessing steps such as TF-IDF and word embedding for effective analysis. Social media platforms have emerged as pivotal battlegrounds in the dissemination of misinformation. The outbreak of the COVID-19 pandemic underscored the urgency of combating the rampant spread of fake news, prompting interventions from global health organizations like the World Health Organization (WHO) to mitigate the associated 'infodemic.' NLP methods are essential for determining the authenticity of written material, enabling sentiment analysis and named entity recognition across diverse linguistic domains.

However, challenges persist in extending these techniques to non-European languages, highlighting the need for further research in linguistic diversity. Machine learning algorithms, particularly those under the umbrella of supervised and unsupervised data mining, offer promising avenues for fake news classification. Decision tree and random forest algorithms excel in capturing nonlinear relationships within datasets, while support vector machines (SVM) provide robust classification capabilities.

In the field of identifying misinformation, the amalgamation of diverse classifiers holds promise for enhancing classification accuracy. However, challenges persist in validating the efficacy of these models across varied datasets, necessitating ongoing research efforts.

III. PLANNED SYSTEM

The system described in the current study is employed to detect misinformation through a range of supervised learning techniques. In this system, diverse machine learning classifiers and combination of these models are applied to differentiate between trustworthy and untrustworthy news sources. Initially, data collection is undertaken to assemble information for training and testing these classifiers. Subsequently, data preprocessing involves conversion of textual information into numerical data. The dataset is then subjected to validation, dividing it into k equal segments where one segment serves as training and the other as testing. Lastly, Evaluation of these models takes place. The simple representation shown below:
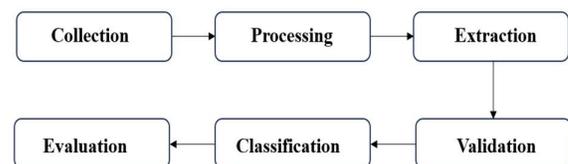

Figure 1: Simple workflow

Additionally, a detailed workflow for the described system is as follows: To effectively employ a Machine Learning (ML) model, a substantial dataset is imperative for training and testing. Utilizing web data extraction techniques, particularly web scraping, data can be extracted from various sources on the internet and structured into a tabular format stored locally or in a database. The process involves employing web crawlers such as 'Scrapy' in conjunction with Python scripts to gather data comprehensively, subsequently

analysing it as per specific requirements. 'Scrapy,' being a Python-based web crawler, aids in retrieving desired data by iteratively scraping it from designated URLs. Refer to Figure 2 for an detailed overview of the workflow.

Once data is amassed, it's segregated into 2 critical sets: training & testing, crucial for assessing model accuracy. The conventional Train/Test methodology involves training algorithms on a large dataset comprising manually examined web pages. Prior to feeding raw content into simulations, certain preprocessing steps are necessary to refine the data. Data preprocessing serves to convert original data into a suitable format for analysis. Given the inherent inaccuracies in real-life data, preprocessing becomes essential to rectify inconsistencies, ensuring optimal performance during subsequent analysis. Techniques like tokenization and stemming are employed as part of the preprocessing phase. The collected material, sourced from websites suspected of disseminating fake news, necessitates transformation into numerical representations before input into a ML system. This transformation is vital for the predictive algorithm, which operates on documents to generate class labels for document classification. Textual data undergoes tokenization to parse out specific terms. Feature selection and extraction methods are then applied to identify relevant features contributing to prediction accuracy. In feature selection, methods like TF-IDF, n-grams and bag-of-words weighting are commonly employed. These techniques are implemented using Python libraries like scikit-learn. The Bag of Words model, a fundamental ML approach for text parsing, disregards word order and focuses solely on word frequency within documents. Implementation choices encompass the Count Vectorizer and the TF-IDF Vectorizer. The Count Vectorizer produces encoded vectors representing the entire vocabulary's length alongside word frequencies within documents. TF-IDF algorithm assigns scores to words based on their frequency, highlighting their importance within documents. TF-IDF Vectorizer facilitates document conversion into tokens, vocabulary learning, inverse document frequency weighting, and supports ciphering new documents.

Following the detailed workflow outlined, the next steps involve training the machine learning model using the segregated datasets, namely the training set and the testing set. This training phase entails applying various classification algorithms

to the training data, leveraging the selected features identified during the preprocessing stage. These classifiers automatically organize or categorize the data into predefined classes, which in this context could include distinguishing between fake and genuine news articles. The model undergoes assessment using the testing dataset to gauge its precision and efficiency in categorizing unfamiliar information. This assessment aids in enhancing the model and adjusting parameters to maximize its performance. Additionally, ongoing monitoring and validation are essential to ensure the model's reliability and adaptability to new data. Through iterative refinement and validation cycles, the machine learning system evolves to effectively combat the spread of misinformation and enhance decision-making processes based on analysed textual data. The detailed representation shown below:
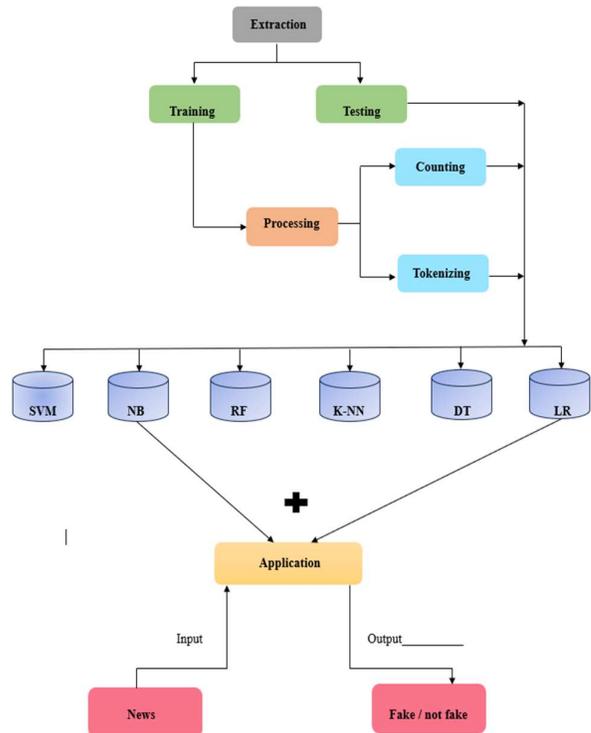

Figure 2: Detailed workflow

IV. MODEL EXECUTION

A. Collection

The data used in this study was acquired from a reliable source, ensuring its reliability and integrity. It consists of a structured format with multiple columns containing information pertinent to the analysis. The dataset encompasses a

substantial number of entries, providing a diverse range of data points for comprehensive investigation. It is worth noting that the dataset has been thoroughly curated to meet the standards of quality and accuracy required for meaningful research.

B. Processing

Text processing involves several essential steps:

a.) *Eliminating extraneous characters*: Initially, the text undergoes a cleansing process where numerical digits, punctuation marks, and unnecessary symbols are removed to enhance the accuracy of analysis.

b.) *Lowercasing uppercase characters*: Subsequently, all uppercase characters are converted to lowercase. This normalization ensures uniformity in text interpretation, preventing discrepancies arising from distinctions between cases, such as "Shop" and "shop".

c.) *Tokenization*: The text is divided into meaningful segments known as tokens. This segmentation assists in understanding by breaking the text into manageable parts. For example, consider the phrase "I enjoy reading." It is broken down into separate tokens such as "I," "enjoy," and "reading."

d.) *Elimination*: Words, which contribute minimal semantic value (i.e., stop words), are excluded from the text. By discarding these superfluous terms, the efficiency of text processing is improved.

e.) *Stemming*: Stemming is the process of reducing words to their root form, ensuring consistency in word representation. For example, words like "running," "runner," and "runs" are all stemmed to the common root "run."

f.) *Rewording:* In the last stage, the tokenized text is reconstructed into a cohesive string format, prepared for additional analysis or presentation, without any plagiarism.

Algorithm	Dataset	F1 Score	Precision	Accuracy	Recall
Logistic Regression (LR)	Large	0.87	0.99	0.87	0.89
Naïve Bayes (NB)	Small	0.99	0.99	0.85	0.95

Figure 3: Comparison of 2 best algorithms

C. Extraction

The current approach involves employing a TF-IDF vectorizer for feature extraction. This method converts textual information into numerical values [22-24]. Following this conversion, the numerical data serves as input for a machine learning classifier to make predictions. The TF-IDF Vectorizer comprises two components:

1.) TF: This indicates the occurrence rate of a word within a particular report or segment of the corpus.

2.) IDF: This measures the frequency of occurrence of a word across different reports or lines within the corpus.

D. Validation

This stage involves employing k-fold cross-validation to partition the dataset into 10 folds.

E. Classification

Various machine learning algorithms are available for classification tasks. This study employs six distinct algorithms:

i. *Logistic Regression (LR):* Logistic regression presents numerous advantages, including its capability for probability modelling, the ability to incorporate features, and adaptability for model updates. Nonetheless, to achieve higher accuracy, logistic regression necessitates a large dataset, whereas Naive Bayes can perform effectively with smaller datasets.

ii. *Naïve Bayes (NB):* Naïve Bayes, an algorithm, makes predictions relying on the probability of specific events occurring. Its operation can be delineated in three sequential steps. Initially, the dataset is transformed into a frequency table. Subsequently, a likelihood table is constructed to aid in forecasting feature probabilities. Finally, the Bayes theorem is employed to compute the probabilities.

iii. *K-Nearest Neighbours (KNN):* The KNN algorithm functions based on the premise that the incoming data point and the existing data points share similarities, thus assigning the new point to the category most akin to those existing ones. This process allows for swift categorization of new data by utilizing the KNN algorithm. While capable of both regression and classification tasks,

Harnessing Machine Learning for Detecting False Information

KNN is predominantly employed for classification conundrums.

iv. *Decision tree (DT):* The concept of decision trees revolves around a hierarchical structure composed of decision nodes, which begin at the apex and progress downward. Unlike methods requiring linear class separation, decision trees accommodate dependent characteristics efficiently while swiftly managing outliers. Moreover, they offer an intuitive interpretation of decisions. Nevertheless, in scenarios characterized by a multitude of sparse features, decision trees tend to overfit, resulting in subpar performance when applied to testing data.

v. *Random forest (RD):* The random forest algorithm builds multiple decision trees, each using a different subset of features. Every decision-tree (TR) generates a single class and subsequently aggregates the votes to improve accuracy using the Random Forest technique. A decision-tree (TR) utilizes a structure resembling a tree to depict its decision-making process, with decisions being made at each node within the tree.

vi. *Support Vector Machine (SVM):* SVM is a robust algorithm employed for classification purposes. It operates by constructing a boundary line within the n-dimensional space, effectively partitioning it into distinct regions to facilitate the classification of new data points. This boundary line, often referred to as a hyperplane, is strategically positioned to maximize the margin between different classes. The algorithm identifies the most pivotal data points, termed support vectors, which play a crucial role in determining the hyperplane.

F. Evaluation

This step is evaluating the model and generate a response whether a news is fake or real.

V. RESULT

After familiarizing ourselves with various metrics utilized within the project, an analysis of the accuracy scores for six classifiers was conducted. Upon examining Figure 4, it becomes evident that combination of the Logistic Regression and Naïve Bayes algorithm exhibited superior performance, whereas the K-Nearest Neighbour algorithm fell short of expectations, demonstrating the poorest performance. Figure 4 presents the accuracy for all six classifiers. Notably, the combination of the Logistic Regression and Naïve Bayes algorithm demonstrated exceptional performance, achieving a remarkable accuracy of 0.98. Conversely, the k-Nearest Neighbour algorithm yielded the lowest accuracy standing at 0.81. Furthermore, Figure 5 juxtaposes the accuracy of our algorithm with that of previous findings. Consequently, our algorithm demonstrates superior performance, yielding exceptional results.
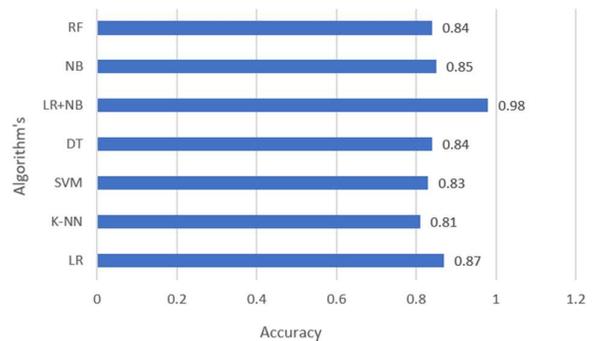

Figure 4: Comparison of accuracy of algorithms

Article	Year	Algorithm used	Accuracy
R. V. L, C. Yimin, and C. N. J (2016)	2016	NLP	0.76
Y. Seo, D. Seo, and C. S. Jeong (2018)	2018	CNN	0.86
"Our findings"	2024	LR+NB	0.98

Figure 5: Comparison with previous findings

VI. CONCLUSION

The dissemination of misinformation masquerading as genuine news creates confusion among the general populace. This phenomenon involves the propagation of inaccurate or deceptive content disguised as factual reporting. While fake news was once primarily associated with traditional print media such as newspapers, its prevalence has surged in contemporary times, largely facilitated by platforms like WhatsApp, Instagram, and Facebook. Various measures are being implemented to mitigate its dissemination; alongside significant research efforts aimed at developing sophisticated tools to address this

issue. Among these initiatives, our proposed system stands out, leveraging reputable news sources to validate user-submitted claims. Through a process of comparison and, our system yields results with high accuracy.

REFERENCES

1. A study by Jain et al. (2019) presents a system utilizing machine learning to detect fake news, showcased at the 2019 International Conference on Issues and Challenges in Intelligent Computing Techniques.
2. Smitha and Bharath (2020) conduct a comparative analysis of various machine learning classifiers for fake news detection, discussed at the 2020 Second International Conference on Inventive Research in Computing Applications.
3. Khurana et al. (2021) evaluate the efficacy of supervised machine learning and feature selection techniques in diagnosing heart disease, presented at the 2021 8th International Conference on Signal Processing and Integrated Networks.
4. Saikh et al. (2019) propose a novel approach to fake news detection, employing deep learning augmented with textual entailment features, detailed in the proceedings of the International Conference on Applications of Natural Language to Information Systems.
5. Manzoor et al. (2019) conduct a systematic review on fake news detection utilizing machine learning approaches, presented at the 2019 3rd International Conference on Trends in Electronics and Informatics.
6. Kapsikar et al. (2021) analyse the collective tagging method for controlling fake news through a branching process analysis, discussed in IEEE Control Systems Letters.
7. Zhou et al. (2019) demonstrate the vulnerability of fake news detection via NLP to adversarial attacks, presented at the ICAART 2019 - Proc. 11th Int. Conf. Agents Artif. Intell.
8. Granskogen and Gulla (2017) explore the use of network data from social media to predict fake news, discussed in CEUR Workshop Proceedings.
9. Garg and Sharma (2020) describe the creation of an annotated corpus for sentiment analysis in code-mixed Hindi-English social network data, published in the Indian Journal of Science and Technology.
10. Lingam et al. (2018) present a method utilizing deep learning for detecting conflicting statements in text, available as a tech report in PeerJ Preprints.
11. Li et al. (2021) propose an Entity-Oriented Multimodal Alignment and Fusion Network for fake news detection, detailed in IEEE Transactions on Multimedia.
12. de Lima Salge and Berente (2017) investigate the ethical behaviour of social bots in their study published in Communications of the ACM.