

Modelling a Machine Learning Framework for Detecting and Classifying News Either As Fake or Real

CHEGE KIMONDO JOHN

Faculty Of Computing And Information Technology In Partial Fulfillment Of The Requirements For The Award Of Master Of Science In Computer Science Of Multimedia University Of Kenya

²Dr. BONFACE NGARI, ³Dr. CPA LEONARD WAKOLI

²Multimedia University of Kenya

³Multimedia University of Kenya

ABSTRACT:- In Kenya, fake news can have a profound impact on public discourse and social stability, as it can be used to spread misinformation and stir up political tension and unrest, which was seen during the 2017 elections. Traditional approaches to detection tend to focus on keyword matches or static lists, which can be less effective in dealing with new patterns of misinformation. This study aims to fill that gap by implementing a machine learning to detect and classify news articles as fake or real, using both textual and contextual features instead of a single type of feature. The contribution of this work lies not in the classification algorithms themselves, which are established techniques, but in combining TF-IDF text vectorization with hand-crafted stylistic features into a single representation and grounding that representation in a purpose-built, manually verified corpus of Kenyan news. A data set of 8,100 news articles was sourced from various Kenyan media and fed into a ETL(Extract Transform Load) pipeline built on PostgreSQL which was then subjected to quality control to produce 6,364 cleaned news articles. Articles were manually fact-checked, assigned a score for source credibility and cross-referenced with trusted fact checking organizations, resulting in a balanced sample of 3,485 real and 2,879 fake articles. TF-IDF vectorization (5,000 features) was joined with eight hand-crafted linguistic indicators, representing feature engineering. This amounted to sentiment polarity, punctuation patterns and capitalization to create a 5,008-dimensional representation. Three classifiers were trained and compared. These were Logistic Regression, Random Forest, and XGBoost, while Logistic regression had the highest accuracy (95.31%), F1-score (0.9478), precision (0.9237), and recall (0.9732) on a test set that had not been used in training the models. The final model was deployed through a REST service implemented in FastAPI, which allows for single and batch predictions, with all predictions logged into PostgreSQL for future drift tracking and an intended model retraining cycle every month. The findings provide a real-world test of the feasibility of a complete and end-to-end system for the detection of fake news in the context of the Kenyan media.

I. INTRODUCTION

1.1 Introduction

Over the last 2 decades, digital connectivity has grown faster and more unprecedented than ever before. It has also fundamentally changed how information is created, shared, consumed and interpreted around the world. Today, 70% of the world's population is actively engaged in online environments, and most people get news and are involved in civic life through social media platforms like Facebook, X (formerly Twitter), WhatsApp, and TikTok. (Allcott & Gentzkow, 2017) This digital revolution has played a central role in making information more accessible. Moreover, it has enabled people to have more platforms to voice their opinions, more opportunities to establish grassroots political organizations and more means to hold powerful institutions to account. Lastly, it has also created fertile ground for the deliberate and rapid dissemination of false or misleading information often dubbed fake news by the people. These platforms are algorithmic and often the algorithm prioritizes sensational emotionally-driven content. This can lead to the spread of fabricated stories faster. (Vosoughi et al., 2018)

Fake news is broadly defined to refer to information that is deliberately fabricated, clearly untrue, or extremely misleading. (Molina et al., 2021) It is often meticulously crafted and designed to resemble legitimate, authoritative and fact-based media in visual form, structure and tone. It is produced with the intent to manipulate

public opinion and/or exploit cognitive biases (Pennycook & Rand, 2021) in order to serve a political, financial, or ideological agenda. But the impact of fake news goes far beyond the realm of misinformation or harmless rumors. Fake news have proved to have a major effect on swaying national elections which may in turn lead to social and political polarization. (Allcott & Gentzkow, 2017) Fake news undermines public confidence in the traditional media and even goes to an extent of even inciting ethnic conflicts as well as real-world violence in the society.

Thus, identifying, monitoring and preventing the propagation of fake news before reaching a large and vulnerable population has become a pressing social need as well as a crucial technical challenge. (Zhou & Zafarani, 2020)

Machine learning is a field that is very promising in tackling this menace of fake news. Machine learning has the potential to carry analysis on a huge amount of both structured and unstructured text. This is performed by extracting complex patterns that exists in the language structure and context in order to distinguish between fabricated and authentic content. (Ahmed et al., 2021) These patterns are usually too nuanced and very complex to manually handle with the raising demands in the current world. To solve this problem this study aims at designing, developing and testing an automated machine learning model that will help classify news as either authentic or fabricated in the Kenyan context. This system relies on a comprehensive end-to-end pipeline. The process will begin with detailed data collection from local sources as well as international sources. Once the data is collected it will be taken through a thorough text pre-processing, intricate feature engineering, repeated model training, then deployed to a user-friendly API.

1.2 Background Information

The political, ethnic and social intricacies of Kenya have had a huge impact on her complicated history with misinformation and digital misinformations. Fake news has become a particular and alarming feature of the Kenyan elections during elections. For instance, in the past three general elections held in Kenya, highly fabricated stories, doctored images and manipulated and fake polling results were shared and distributed in the social media. (Maweu, 2019) The coordinated spread of fake news across the various media resulted in a significant impacts on the voters' perceptions. Moreover, sharing of this information led to outbreak of confrontations in various areas in the country and this led to rise in political tensions in the country. (Madungu, 2022) This was a period of profound systemic weakness as Kenyans were increasingly consuming their news on digital platforms and social media channels instead of traditional print and broadcast outlets. Due to this a lot of unverifiable, misleading and harmful messages proliferated unchecked on the internet.

The global battle against fake news has generated an ever-increasing amount of academic and industrial research. Research in this area have successfully used different machine learning methods to support the classification problem. Algorithms like Support Vector Machines (SVM), sophisticated Natural Language Processing (NLP) systems, and keyword-based sentiment lexicons, as well as amalgamation systems (Aslam et al., 2021), have been effectively employed to separate actual information from fake news. (Choudhary et al., 2021) The accuracy of these studies varies widely, from about 79% to more than 90%. However, these results are strongly dependent on the language used for the data set, the linguistic features of the text and the algorithmic methods employed. Nonetheless, much of this previous work has significant drawbacks. A lot of models are only based on a single predictive indicator. For instance, they would just analyse the headline, looking at the domain of the source, or using a set-up keyword list that can easily be circumvented. Others are trained on datasets that focus only on breaking information, while others are trained on current news. Thus they often overlooks older, evergreen misinformation, often resurfacing, or don't provide enough mathematical explanation of their underlying algorithms to facilitate academic reproduction and trust by users. (Zhou & Zafarani, 2020)

In addition, most current detection systems are largely developed, trained and tested from massive amounts of data from Western contexts (e.g., American or European news sources). (Ahmed et al., 2021) There is thus a large and important gap in the computational modelling literature for models specifically designed to the unique language characteristics, multiple media environment, and particular socio-political dynamics of Kenya. With misinformation not operating in a cultural vacuum, its effects and dissemination are strongly influenced by political history, news media trust, and the use of multilingual languages such as English, Swahili, and Sheng (Madrid-Morales et al., 2021). Therefore, a detection model that specifically targets the news sources and consumption patterns in Kenya is essential to tackle the misinformation issue in this particular context.

This study directly addresses the critical need by developing a specialized fake news detection pipeline. It is a fusion of the linguistic and stylistic features. It includes excessive punctuation patterns, irregular capitalization usage, and sentiment polarity, with the Term Frequency-Inverse Document Frequency (TF-IDF) text vectorization. This feature set is trained and thoroughly tested on a newly collected corpus of news articles from Kenya. It is then released as an operationalized Application Programming Interface (API) for real-time classification. This helps in continuous performance monitoring to counter concept drift.

1.3 Problem Statement

Lack of proper information has become an acute problem all over the world. With fake news on the rise and its effects on public opinion, economic situations, and social unity it has become a persistent menace. This is especially true in Kenya where there has been a lack of ability to identify and correct misinformation. This has regularly led to high political polarization and civil violence during key moments in the country like general elections. In 2017 and 2022 elections, false information was widely and systematically spread on digital platforms during the election campaigns in Kenya. (Madung, 2022) The messages had a clear impact on how Kenyans voted, undermined voter turnout in some parts of the country and exacerbated ethnic tensions. This therefore, reinforces the critical need for effective, automated approaches to address digital misinformation.

Current fake news detection techniques, however, are far from adequate if applied in the local context. They often use a simple keyword search and employ fixed, precompiled data sets. (Shu et al., 2017) These existing detection methods can't match the highly advanced, constantly changing strategies employed by today's disinformation campaigns. These traditional methods are not as contextually aware as they need to be in order to identify sarcasm, subtle framing or terms that are new and deceptive. (Zhou & Zafarani, 2020) It is also important to be aware that even "true" news can contribute to tensions in society. Selective news reporting, ideological bias, sensationalism and framing can influence how the public sees news with the intention of deliberately escalating conflicts and incite violence.

1.4 Research Objectives

1.4.1 General objective

The general objective of this research is to develop a robust machine learning model that effectively detects and classifies fake news in real-time thus enhancing integrity of information shared on social media platforms

1.4.2 Specific Objectives

- i) To Assess the existence of fake News on Social Media platforms in Kenya
- ii) To analyze existing models used to detect and classify News as fake or genuine
- iii) To design a machine learning model that can be used to detect and classify news as fake.
- iv) To test and validate the machine learning model developed.

1.5 Research Questions

- i) How do we assess the existence of fake News on Social Media platforms in Kenya?
- ii) How do we analyze existing models used to detect and classify News as fake or genuine?
- iii) How do we design a machine learning model that can be used to detect and classify news as fake?
- iv) How do we test and validate the machine learning model developed in this study?
- v) Does combining TF-IDF vectorization with hand-crafted stylistic features, applied to a purpose-built Kenyan news corpus, offer a measurable advantage over a single-signal feature representation, and can this combined representation be extended with a Kenyan-context-aware linguistic feature set in future work to strengthen that advantage further?

1.6 Significance of the Study

The findings of this study on fake news detection by machine learning have a number of significant and long-term implications for both research and applications. First, it will greatly improve our empirical knowledge of the social and behavioural processes. Further, it will enlighten us on factors that affect the rapid spread of misinformation in Kenya. Especially during moments of heightened sensitivity and criticality such as national elections. A systematic identification of the linguistic and structural features of fake news can help key stakeholders, such as the Independent Electoral and Boundaries Commission (IEBC), media councils, and tech regulators, come up with focused and evidence-based approaches to counter fake news and ensure the sanctity of public discourse.

Secondly, the research will critically examine the effectiveness of different machine learning algorithms in the context of detecting fake news in a developing country context. This will be of significant value for local media outlets, fact-checking NGO's or technology developers when choosing the most suited computational tools and techniques for real-time detection. The ultimate goal of this will be making information shared on social media more reliable and trustworthy.

Further, the findings of this research will play a crucial role in providing insights into the overall long term effects of fake news on public trust and also in maintaining social stability in the nation. Through this study, the government and other key players in the public will be able to comprehend the influence that fake and inaccurate information have on social dynamics. This will force them to take the necessary measures in

promoting digital literacy and also come up with ways to combat sharing of fake news online. This in turn will lead to creation of an informed, resilient and an engaging nation.

If the government stakeholders and the education key player comprehend how inaccurate information influences social dynamics, then they will take the necessary measures to promote digital literacy. This will help in creating an informed, resilient and an engaging society.

Finally, this study will not only have implications for Kenya, but will also be a baseline study and scalable model. It can be used as a baseline study for developing nations in Sub Saharan Africa and other nations around the world who are experiencing similar issues with digital misinformation. The findings from this research can help make a real difference in combating misinformation and advancing the integrity of digital information systems around the world by providing insights into strategies that can be implemented in practice. To sum up, this research has the goal of offering technical knowledge which can serve as a guide for the stakeholders to the problem of fake news. While fake news poses a global challenge, it particularly affects Kenya because of the influence of historical events which have often resulted in deadly conflicts and loss of life. This is directly attributed to unchecked dissemination of fake news, such as the post-2007 election violence and the tensions experienced during the 2017 election.

1.7 Justification of the Study

There are a number of reasons why the study of detecting fake news with the help of machine learning is so important and topical. First, the outpouring of misinformation is fast and unchecked in Kenya. It has been well documented as having a serious impact on democratic processes, national security, and social cohesion especially during election time. (Maweu, 2019) With digital platforms and smartphones now being widely used by a majority of Kenyans, even among those that consume traditional broadcast media. (Portland Communications & GeoPoll, 2017) There is an urgent and undeniable need to find a way to effectively, at scale, identify and counter fake news. In this study, we seek to solve this urgent national problem. This will be managed by introducing a systematic, efficient method to detect misinformation and consequently improve the integrity of information disseminated over the internet.

Secondly, there is the need for more advanced and scalable technological solutions as social media is now used as a daily source of news. The sheer volume of digital information produced and disseminated everyday makes manual fact-checking using humans simply inadequate, time and resource consuming. This research aims to highlight the power of machine learning to predict the effectiveness of different algorithms in identifying fake news in non-Western content. (Ahmed et al., 2021) It provides a technical know-how that can be practically used in the real world and high-stakes situations.

Moreover, this study fits into the national government agenda towards digital transformation. The government aspires that the citizens are digitally enlightened and they can engage more in public discourses regarding digital transformation in the country. As the government and various private players gear up for integration of technology in governance, civic participation, and communication, comprehending how to effectively counter digital misinformation will be crucial. The results of this research will play a role in shaping policy and regulations for responsible sharing of information among citizens and responsible digital use. (Government of Kenya, 2019)

Lastly, the findings of this study and the open-source frameworks that has been developed during this study will not only be useful for the country but also a replicable model can be adopted in other developing nations that face the same problems of fake news. By scaling this study, a more reliable and secure information systems landscape can be built. Furthermore, this research paper will contribute to the development of a more informed and democratic society by contribution to the global knowledge exchange of best practices and algorithmic insights.

1.8 Limitations and Scope of the Study

There are quite a number of limitation in this study of fake news detection using machine learning that ought to be clearly outlined here. First, the geographical and contextual concentration is largely on the Kenyan context. The study focuses on the effects of fake news, linguistic characteristics and propagation of such fake news on public opinion and social stability during critical events like elections. Therefore, although the general machine learning structure might be transferable, the results, features, weights and model parameters may not be directly transferable to other countries or regions, which have different socio-political circumstances, languages or media regulations. (Madrid-Morales et al., 2021)

Second, the research will investigate, train, and compare different machine learning algorithms and the effectiveness of each of them in detecting fake news. However, it will not go through all the deep mathematical and computational problems of optimizing each one from scratch. The fact that it is a limitation is likely to mean that the study may not offer in-depth and detailed information on the subtle variations in algorithm performance

at the micro-level when applied to extremely diverse, global data sets, and may instead offer practical and comparative accuracy.

Furthermore, the study mainly focuses on the public social media and digital news portals as the primary platforms where fake news is spread. It will not attempt to systematically discuss and technically analyse the other potential sources of misinformation private messaging platforms such as WhatsApp or closed Telegram groups. These private messaging platforms pose a very high technical challenge for web scraping but they are very significant in the local transmission of fake news.

During the research, another practical limitation was observing that there might not be enough human respondents and subject-matter experts having hands-on experience in both machine learning engineering and the sociological research of fake news detection. There were challenges with accessing key audiences for in-depth discussions with government officials, senior media practitioners and technology developers. This is because everyone had an interest in, or was able to provide, their internal insights and experiences of moderation strategies.

Finally, the study made extensive use of available and scraped data, and though it was thoroughly cleaned, it did not capture the full absolute volume and variety of fake news that were being disseminated in Kenya. The accuracy and robustness of the machine learning model developed was critically dependent on the available datasets quality, size and also demographic diversity that was used in testing the machine learning model. While, the study notes this as some of the weaknesses, the study pursuit to deliver an actionable and highly valuable insights on detecting fake news in Kenya that would improve the academic and practical comprehension of digital misinformation has been greatly achieved. These challenges were addressed thoroughly by trying to obtain as much information from various secondary sources, using data augmentation techniques, and looking at existing literature.

II. LITERATURE REVIEW

2.1 Introduction

This chapter places the present research in a systematic context of the existing and rapidly growing knowledge regarding the automated detection of fake news, computational linguistics and the application of machine learning architectures in mitigating misinformation. (Zhou & Zafarani, 2020) It starts with an overview of the academic debates around digital misinformation, examining the socio-technical, psychological and algorithmic mechanisms that enable misinformation to rapidly circulate (and proliferate) through networked platforms. The review critically examines the contribution made by machine learning in the modern large-scale campaign against information manipulation, and specifically the unique and essential Kenyan digital media ecology. For Kenya, with a very active mobile internet penetration rate, multilingual digital communication patterns (English, Swahili, and Sheng), and the documented impacts that coordinated inauthentic behavior has on the country during the national electoral cycle, localized computational approaches are not only desirable but necessary to the empirical investigation. The chapter then moves to a detailed, critical analysis of related research, presenting a comparative analysis of five fundamental publications that have taken a different approach using various machine learning techniques to the multi-dimensional, complex problem of fake news classification. These include Support Vector Machines, Natural Language Processing pipelines, keyword-based lexical matching, linear regression frameworks, and Long Short-Term Memory recurrent neural networks. Each study that is provided here has been carefully considered and their specific approaches to data curation, algorithmic realisations and their reported accuracy as well as their practical applicability has been reported. This is to ensure that the studies meet the criterion required to back up a new study.

In this chapter, a summary of the literature is provided. It includes the the main methodological trends, the recurrent methodological difficulties (Choraś et al., 2021) and the general conclusions and implications for future research. This synthesis highlights common the glaring gaps that exist between the current researches and this research. This section also includes the continued use of single-signal detection approaches, the periodic re-emergence of historical misinformation, and a significant lack of algorithmic transparency and explainability. It further focuses on the geographic bias of western and English-dominated datasets. The chapter also delves into the various factors influencing the organizational, institutional and societal adoption of fake news detection models. This spans from the swift technological developments, the increasing demand by the public for digital credibility, to the strict regulations imposed on them and the need for digital credibility for risk management purposes by companies. The chapter then summarizes the major tools, platforms, and technologies stacks (programming languages, machine learning frameworks, natural language processing libraries, relational databases, and cloud-native deployments) that are used in the specialized field. It also takes into account the practical, ethical, computational and resource-based issues that organizations are likely to encounter in the deployment of such predictive models. This include but not limited to stringent data quality requirements, inherent models' algorithmic biases, latency requirements and the costs of regular models' maintenance.

Lastly, the chapter describes in detail a conceptual model that thoroughly outlines the elements of a highly effective, production-ready fake news detection system. This framework clearly identifies the input variables, the sequential processing and feature extraction phases. It also includes the architecture selection of the model, the classification mechanism of the outputs, the continuous feedback loop and the longitudinal impact assessment measures. This will help in providing a solid structural and theoretical basis for the methodology presented in Chapter Three. Overall, the current review makes it clear, as a group of scholars, that there is a need for a model that can be adopted in the Kenyan context. The model ought to integrate multiple, diverse feature types including lexical, syntactic, semantic and stylistic features (Albahar, 2021) as opposed to a single, easily manipulated detection signal or a context-blind algorithmic pipeline.

2.2 Related Literature

In this extensive paper, Julio C. S. Reis et al. (2019) discuss and assess the various different computational techniques that are used in identifying fake news. They give special attention on classification approaches based on metadata. The researchers emphasize on the significance of structural characteristics such as the news title as well as the source domain of a news site in verifying the credibility of the article. In this article, the researchers claim that malicious actors tend to follow a predictable pattern in developing news titles as well as the sign for domain systems. In this study, Julio C. S. Reis et al. (2019) use a predictive model (Support Vector Machine (SVM) classifier (Cortes & Vapnik, 1995)) that is used widely in classifying problems in a binary manner. The binary factors in their research being the model and also the basic lexical components. Their algorithm achieves an 87% accuracy on the test data which is quite impressive. But their approach is criticised more closely for being too subjective. Researchers argue that comparative baselines and cross validation methods are inappropriate and fail to allow for the universal or generalizable conclusions that are required. This limitation is largely due to the fact that more advanced fake news and coordinated disinformation campaigns are unlikely to be identified just from the title and the source of the information. This is because today's cybercriminals are adept at using new registrations of domain names similar to trusted ones, sophisticated spoofing of trusted domain names and using emotionally optimized, clickable headlines that are designed to evade the surface-level metadata filters. This results in a low robustness with changing adversarial tactics because of shallow features, but the SVM is still computationally efficient.

Adrian M.P. et al. (2019) recognise that detecting fake news at speed is a serious and increasing problem and have created a predictive model which focuses on detecting fake news early in the dissemination process, prior to viral amplification. They use a sophisticated Natural Language Processing (NLP) algorithm that makes it possible to robustly and accurately capture both syntactic and semantic differences in language between authentic journalism and fabricated journalism. The authors generate a dataset with only breaking and current news, and extract the training and testing sets that are necessary to perform supervised machine learning, paying special attention to the relevance of the time. They claim to be 79% accurate. (Braşoveanu & Andonie, 2019) This is generally considered to be too low for applications dealing with content moderation in real time when user-generated content is flowing on social networks. In a case where a false positive can result in the suppression of legitimate content, while a false negative can lead to the propagation of harmful ideas. The main reason of this less accurate data is that the study only investigates current news. There is well known empirical evidence that fake news often is generated from a much older, evergreen, and widely spread content that is deliberately repurposed to provoke new outrage or to intervene and influence current news reporting. This ignores a crucial lack in real time detection systems, which do not take into account past misinformation patterns.

In the scope of computational content monitoring, William Yang Wang's (2017) research relies greatly on keyword identification and lexical analysis for identifying fake news. The use of this kind of approach is relevant and very common in the fields of regulation and surveillance, such as in the context of large-scale frameworks of digital communication monitoring. The author decomposes news articles into a set of words and n-grams, and stores them in a structured lexicon array, which associates term frequencies with indicators of credibility. He builds a large database of both old and new news and then divides it into training and testing sets, using a typical ratio of 70:30. This splitting is to allow for an effective evaluation of the model. The model takes into consideration several factors including media coverage, publication timing, semantics of the title and credibility of the source. Finally, the model is designed to make a prediction using a decision tree algorithm which has interpretability and a rule-based output. The algorithm is reported to be accurate with over 92%. (Wang, 2017) However, its dependence on an exhaustive keyword search and static lexical matching can lead to a significantly more complex algorithm, particularly time complexity and space complexity. Moreover, latency issues may arise when applied to high traffic social media sites for practical, real-time use. Finally, keyword-based systems are easily fooled by adversarial evasion, by replacing targeted words with their synonyms or codewords.

According to Costin Busioc et al. (2020), the spreading of fake news on social media represents an important systemic phenomenon that affects the integrity of democracy and public health. The researchers use an unusual and statistically underpowered algorithm (linear regression) to help them determine whether a news item is fake or not. They build their own fake news/confirmed fake news, true news/confirmed true news dataset to train the algorithm. The authors have decided to use a specific composition of the news: 65% true news and 35% false news, which is divided into training and testing sets in an 80:20 ratio to simulate the distribution skew found in the real world. They claim that they are 91% accurate on their test set (Busioc et al., 2020), which they tested under controlled conditions. The paper is, however, rather shallow in its methodological approach and analysis. It would have benefited from a much more thorough mathematical modelling of the formulation of the algorithm, the thresholding process used to transform continuous outputs into binary output and the dealing with class imbalance for academic reproducibility. Moreover, linear regression overlooks the non-linear regression relationship that exists in linguistic and stylistic features. These factors thus reduce the ability of the model to capture complex patterns that are used in deceiving or manipulating people views.

Alim Al Ayub Ahmed (2021) conducts an extensive and comprehensive study on fake news detection. He leverages on the contextual and metadata features such as article title, publication time, domain of the publication and geographical location metadata. The author collects a large, well-balanced dataset of 10,000 news items: 8,000 are true and 2,000 are fake, with the former and latter being evenly distributed across the entire set. Besides, the author classifies the news into different thematic classes namely religious news, political news, social news, criminal news and ritual news which can be used to tune and bias the model within each domain. The author reports an accuracy of 89% (Ahmed et al., 2021) after implementing sequential dependency and long-range contextual dependency in the text with a Long Short-Term Memory (LSTM) neural network (Hochreiter & Schmidhuber, 1997). However, the study doesn't really consider the importance of the time relevance and present information cycles. These are often the most important factors that contribute to the viral spread of misinformation fast. Additionally, LSTMs are good at modeling sequences, but the study does not sufficiently tackle the computational burden, training time or lack of interpretability of recurrent architectures, limiting its applicability

2.3 Empirical Review of Related literature

The academic and technical publications on detecting fake news have changed a lot in the last decade. It mirrors the ever-increasing and pressing need for combating the omnipresent fake news in the modern digital era. This section summarizes the key empirical findings. It also focuses on the methodological trends and the empirical results of some of the foundational studies that have been carried out in this field. The empirical review focuses on both the different computational methodologies used and also the inherent technical and contextual difficulties faced and the implications for future research paths. Initially, many researchers attempted to create models that were largely rule-based and manual heuristics. However, this approach became limited in scope and speed as the volumes of data to be analyzed increased and also as the speed of social media dissemination rose with a high speed. To cope with the rising volumes of information and the rising speed at which social media use has been on the rise, researchers ended up adopting machine learning algorithms. These algorithms added statistics rigor to the field of research. However, they were limited by the features used and the contexts they were applied to. (Ahmed et al., 2021) Currently, more developments have been made in this field and more models have been built. These models however, rely heavily on the foundations of deep learning, transformer architectures and propagation analysis on graphs. (Varma et al., 2021) However, these approaches can be computationally expensive. Moreover, these analysis are inclined geographically to Western articles. The review brings together all these insights with the aim of coming up with a solid automated machine learning model that will classify the news as fake or real. Moreover, the empirical synthesis shows a clear trend that models trained on homogeneous, Western media systems show a sharp drop in performance in Asian, African and Latin American media systems. It highlights the importance of data sets that reflect the local contexts of the African, Asian and Latin American media system, and feature engineering that is culturally informed. (Mishra et al., 2022) This gap directly contributes to the methodological options and strategies of data curation taken in the present study.

2.4 Methodologies Employed

This work used a robust multi-stage approach that consisted of data collection, thorough preprocessing, repeated model training, and model production deployment. The data was collected through a multi-source web scraping and API-based method from BBC News (Kenya section), Standard Media, NTV Kenya, Citizen TV, Daily Nation, Twitter/X Kenya news feed, and existing Kaggle fake news datasets which were carefully adapted, linguistically filtered and localized to the Kenyan context. These tools were used to extract the HTML data, such as BeautifulSoup and Scrapy for efficient, asynchronous HTML scraping, and Selenium for complex, JavaScript-rendered content which depends on client-side dynamic loading, and also to access the Twitter/X

API v2 in a safe and rate-compliant manner using tweepy. Throughout the process, a total of 10,547 article were collected. The process was both rigorous and ethically sound. The data collected met the set target of collecting between 8000-12000 articles for the study. This was an impressive rate of 87.89%

A robust Data Engineering Pipeline was carefully engineered around the complete data life cycle. It is ACID compliant, supports efficient queries and scales up for large datasets. Initially extracted articles were stored in a raw_news table. It contained separate fields for 'headline', 'body', 'source domain', 'canonical URL', 'extraction time', 'publication time' and 'raw classified'. ETL (Extract, Transform, Load) script with python component logic was used to process the extraction validations, extract data, load data in a systematic way, and also to stage the data for further processing. It was scheduled via the cron job or through areflow for dependency management and failure alerts. Data cleaning included a series of text normalization, HTML tag removal, special character removal, unicode normalization, and strict article length exclusion of less than 50 characters to cut out noise, boilerplate and non-article content. Of this, a cleaned, analytically viable data set of 9,726 articles (92.3% of the raw data) was obtained after deduplication using SQL window functions, content hashing, and dropping of incomplete and corrupted records with missing or incomplete metadata.

Labeling adopted three complementary and cross-validated approaches to achieve high ground-truth accuracy and reduce annotator bias. This include manual labeling, and using a range of available historical source credibility scores and editorial track records. Lastly, a systematic cross-referencing with established, independent fact-checking organizations and fact-checking debunking databases was used. (International Fact-Checking Network, 2022) The result of this multi-tiered, consensus-based approach was the formation of a dataset of almost equal real articles (54.2%) and fake articles (45.8%), with the articles carefully stratified and preventing class imbalance from skewing model optimization.

This was done by feature engineering, using TF-IDF (Term Frequency-Inverse Document Frequency) text vectorization (Salton & Buckley, 1988) with the top 5,000 features and English stop words removed .Rare terms were filtered, as well as eight hand-crafted linguistic and stylistic features aimed at capturing deception markers. These encompassed Headline and Body-length in characters and words, the total word count, the percentage of capitalisation in the text, the number of punctuation marks presented in the text (particularly exclamation marks and question marks), the number of embedded URLs, and sentiment polarity and subjectivity scores calculated using the TextBlob library, yielding a highly descriptive hybrid feature matrix of 5,008 dimensions, which is sparsely lexical and densely stylistic.

The classification model and TF-IDF vectorizer were serialized using joblib, with efficient binary persistence and operationalized using a high-performance FastAPI REST service. The service was containerized in a multi-stage build to greatly reduce the attack surface, image size, and deployment options were carefully considered for enterprise orchestration on AWS ECS. This is a serverless auto-scaling on Google Cloud Run, rapid developer iteration on Heroku, and on-premise, low-latency deployment on Docker Compose. All predictions, confidence scores and input hashes were directly logged to PostgreSQL to enable structured monthly retraining methodology, statistical drift detection and monitoring of performance degradation for the continued relevance, calibration and accuracy of the model in the presence of inevitable misinformation tactics, linguistic trends and platform algorithms.

2.5 Implications for Future Research

The results of this implementation have broad and deep implications for future studies on the detection of fake news, especially in the context of the Kenyan media landscape and beyond in Africa. First, the combined use of TF-IDF and hand-crafted features (Hakak et al., 2021) resulted in good and empirically supported results (95.31% accuracy). However, it is important to note that the problem of whether there is any added value of including source credibility scoring, social network propagation patterns, user account metadata, or multimodal features such as embedded image size, reverse image search results, or deepfake detection metrics remains open. The literature search pointed to the contextual and multimodal dimensions as potentially unexplored but promising dimensions. These were intentionally excluded from the present pipeline as they were regarded as too high-level to be computationally feasible and to focus on linguistic signals. Future studies should empirically explore the marginal value of these diverse and complex features, relative to the additional engineering complexity, latency and data acquisition costs they introduce.

Second, the process of manual review, source credibility heuristics, and fact-checker cross-referencing used to label the new dataset also has its share of subjectivity, cognitive bias and human errors that are common in most ground-truth fake news datasets. (Choraś et al., 2021) Future work should closely examine inter-rater reliability with multiple independent labelers. It should apply Cohen's Kappa or Fleiss' Kappa statistics, and consider semi-supervised or weak-supervision methods, including data programming, Snorkel frameworks, or heuristic labeling functions, to start scaling the labeling process more. This would ensure the labelling is rigorously, consistently, and cost-effectively carried out on the datasets as datasets expand beyond the current 9,726-article scale to hundreds of thousands.

Third, since the model is trained in a snapshot of the training set, but retrained every month. There are key questions about the rate of performance drop between retraining epochs if the model is used in the wild. Especially with the targeted, rapid evolution of misinformation techniques, adversarial prompt engineering, and platform algorithm changes. Empirical measurement of this concept drift degradation curve (Gama et al., 2014) is needed in future studies. This is to ensure there is development of automated performance threshold alerts.

Fourth, since the dataset and model have been developed specifically for Kenyan news sources linguistic issues might arise. Linguistic nuances, Sheng code switching patterns and the local reporting style must further limit the generalizability and transferability of the approach to other African or developing-country contexts. (Madrid-Morales et al., 2021) Further studies are needed to evaluate the cross-country transferability, domain adaptation approaches, and multilingual fine-tuning approaches. As there are linguistically meaningful and significant differences in baseline media trust dynamics, historical political tensions, and specific misinformation practices across geographical regions and digital ecosystems.

Lastly, although the deployed API can make real-time predictions, it lacks a closed and automated loop. In which the correction by a human verifier, override by a fact-checker, or a user report is systematically and automatically fed back into the retraining data set. Going forward, the feedback mechanism needs to be formalized, possibly by using active learning paradigms, human in the loop validation queues, or reinforcement learning from human feedback (RLHF). This will enable the model to continuously and automatically learn from its own deployment errors, decrease false positive rates over time, and adjust to new deception patterns (Gama et al., 2014) that have not been seen in independent data sources but are new to the model, without the need for retraining on a static corpus.

2.6 Factors Driving the Adoption of Fake News Detection Models

Several societal, technological, regulatory and economic factors have a complicated impact on the broad and rapid use of machine learning models for fake news detection:

- i) **The rapid proliferation of fake news-** This is often observed in highly engaging algorithmically optimised social media platforms that has led to increased global awareness of its potential negatives effects. This effects include misleading people, influencing election results (Allcott & Gentzkow, 2017), fueling ethnic violence and causing social unrest, in ways that are fast and unregulated. This growing need does not leave organizations, civil society organisations, elections commissions and governments sitting on the sidelines waiting for the digital misinformation to take off. All involved parties ought to join hands and actively look for, fund and implement effective and scalable computational solutions before the misinformation achieves critical mass.
- ii) **Technological Advancements-** The speed at which highly sophisticated machine learning algorithms, language models with transformer blocks, and advanced natural language processing techniques and cloud-native compute infrastructure have advanced it has made it possible to compute on huge amounts of unstructured, real-time data efficiently. Such continuous innovations help to identify fake news with a much higher degree of accuracy, providing context and nuance, and are encouraging more stakeholders in the media, tech and government space to deploy these cutting-edge technologies at scale. (Islam et al., 2020)
- iii) **Demand for credibility is growing louder:** Everyday consumers are more aware of the ubiquity of misinformation, psychological tricks by those spreading it, and its risks to society. Therefore, the demand that news be credible, verifiable and clearly sourced is on the rise. (Adjin-Tettey, 2022) Media companies, publishing platforms and social networks that have an effective detection model that is understandable and clearly explained can greatly improve their institutional reputation. Moreover, these organizations can keep their audiences and gain the confidence of their increasingly cynical audience.
- iv) **Regulatory Pressures:** Governments, national and international regulatory bodies are increasingly becoming aware of the critical importance of combating fake news for the security of the nation, public health, and democracy. This has created a strong incentive and even legal obligation to implement automated detection models. Transparency in content moderation and algorithmic accountability will guarantee that the integrity of information shared with the public. Moreover, the government should propose law and acts that have been successfully completed in other nations such as EU Digital Services Act, the UK Online Safety Act and the region's data protection laws. (Government of Kenya, 2019)
- v) **Social Responsibility:** The active and proactive response to combating fake news is considered by many technology corporations, media conglomerates and platform operators, to be part of their corporate social responsibility (CSR) and environmental, social and governance (ESG) requirements. (Tully et al., 2022) They have an opportunity to help build a resilient society, support vulnerable populations, and financially or open-source detection models to help create a more informed society. This would help protect electoral integrity and inter-community relations from the harmful effects of misinformation.
- vi) **The exponential growth of publicly available,** large-scale, fact-checked, annotated social media corpora, historical news archives, and other types of data. This has made in-depth training, validation, and

benchmarking of sophisticated machine learning systems far easier. (Khan et al., 2021) The multitude of different data which is labeled helps the creation of much more sophisticated, generic and context-aware detection systems which can be adapted for a specific region or language.

- vii) **Collaboration and Knowledge Sharing:** There is the need for collaboration among various stakeholders in the nation. This will allow for parties to share knowledge, exchange ideas on the best practices to fight fake news and also come up with models and innovative means that will help combat the spread of fake news in the country. This is a delicate and an interdisciplinary field that requires the stakeholders to collaborate. This will in turn promote rapid and standardized implementation of effective models at the sectoral, geographic and regulatory level, speeding up the field's maturation.

2.7 Tools and Platforms for Fake News Detection Models

Extracting data, creating, training and validating the fake news detection model is a complex system. The model requires a set of complex and interoperable set of tools and platforms from the first stage(data collection) to the last stage(deployment to API). This section will address the critical technological elements that were used in this study.

2.7.1 Programming Languages

The language chosen for this research was Python due to its large, mature and very active data manipulation, machine learning and natural language processing libraries (Pedregosa et al., 2011) and its specialized libraries. These include

- i) Pandas and NumPy for handling data with high performance, matrix operations, and vectorized calculations
 - ii) Scikit-learn for classical machine learning algorithms, model evaluation and pipeline construction (Pedregosa et al., 2011)
 - iii) TextBlob for quick and rule-based sentiment analysis, made the complex feature engineering, hyperparameter tuning, model training stages described in the methodology a breeze. (Hutto & Gilbert, 2014)
- The ease of Python to read, rich package management through pip and conda, and the community around Python, helped to further speed up development cycles and debugging processes

2.7.2 Web Scraping and Data Extraction Tools

BeautifulSoup and Scrapy were used to parse and extract static HTML data from major news websites to access these sites using multiple sources to collect a representative dataset for the area of interest. Selenium was used to support complex pages containing a lot of JavaScript code, and also those which use client-side dynamic loading, infinite scrolling, or anti-bot protection, which requires the use of a browser-automator. Requests handled standard, low-effort HTTP calls that supported session management and proxy rotation, and tweepy delivered secure, rate-compliant, programmatic access to the Twitter/X API v2, making it easy to collect essential data for social media, such as tweet texts, author metadata, and engagement metrics. These tools, combined, were able to successfully and ethically extract 10,547 news articles from a variety of high-traffic news sources such as BBC news, Standard Media, NTV Kenya, Citizen TV and Daily Nation. (Cheruiyot & Ferrer-Conill, 2018)

2.7.3 Database and Data Engineering Tools

The primary relational database management system chosen is PostgreSQL due its robustness, horizontal scalability, strict ACID compliance and advanced capabilities for full-text search indexing, complex analytical queries, and storing JSONB type data. These capabilities were vital to the data pipeline that handles the huge raw_news and cleaned_news tables mentioned in the data pipeline, allowing for deduplication, metadata filtering and historical versions. ETL (Extract, Transform, Load) pipeline was developed by utilizing psycopg2 and SQLAlchemy to effectively connect the database. This ensured seamless data connectivity. Seamless schema management with Apache Airflow was used to make the data ingestion, failure retry logic and alerting processes reliable and automated.

2.7.4 Machine Learning Frameworks and Libraries

This study specifically used the classic machine learning implementations from scikit-learn, Logistic Regression and Random Forest, as well as the dedicated XGBoost library to use the highly optimized, parallelized gradient-boosted classification. (Chen & Guestrin, 2016) This was because the study was highly feature-engineered and structured. The complex vectorization of the text data was performed by the TfidfVectorizer from scikit-learn, which also performs sublinear TF scaling and IDF smoothing. Joblib was used to efficiently serialize, compress, and persist the trained model and vectorizer for low-latency, memory efficient deployment.

2.7.5 API Development and Containerization Tools

FastAPI was used to build the RESTful API that operationalized the trained classifier, chosen for its inference speed, native async support, automatic API documentation, and integration with Pydantic for data validation. The service was containerized with Docker using a multi-stage build to keep the runtime image small and to run consistently across build, staging and production environments.

2.7.6 Cloud and Deployment Platforms

Because the containerized API is portable, several deployment targets were considered rather than tied to one vendor: serverless options such as Google Cloud Run for intermittent, cost-sensitive traffic; Heroku for rapid prototyping; Docker Compose for institutions needing a private, low-latency network; and Amazon Web Services (ECS) for production workloads requiring enterprise-grade, load-balanced scaling. The choice among these is an infrastructure decision to be made per deployment context, based on expected traffic, latency requirements and budget, rather than a finding of this study.

2.7.7 Monitoring Tools

All predictions, input hashes, confidence scores and system latencies were put right on top of PostgreSQL. This enabled to conduct strict statistical drift detection, analyze feature distributions, and track system drift with time.

By doing so, the database itself becomes the monitoring tool, keeping the overall solution simple to install, configure and maintain and cheaper than a third-party monitoring platform. Prediction distributions, class priors, and confidence calibrations were routinely compared to historical baselines through scheduled SQL aggregation queries. Accuracy and confidence thresholds were also established to send alerts and to automatically start retraining pipelines when performance fell below acceptable levels.

In general, it's a pragmatic, production-ready technology architecture that intentionally emphasizes classical machine learning, relational databases, and containers over more distributed and complex big-data architectures and deep learning platforms. It's a design that fits this study's scope and latency constraints, interpretability requirements, and resource constraints and can still be deployed in real-time, scaled horizontally, and continuously monitored in performance.

2.8 Challenges in Adopting Fake News Detection Models (Alghamdi et al., 2022)

In the process of developing, training, validating and implementing the fake news detection model, a number of crucial technical, ethical and operational issues were encountered. However, these issues were dealt with in a systematic manner through the use of iterative engineering and methodological improvements. Some of these and how they were dealt with are outlined below.

- i) One of the biggest issue that was faced in this study was selecting historical and real-time news articles. To solve this issue, the articles were carefully selected to ensure that a balanced, representative and high quality dataset was compiled. However, in the process another issue rose. This was the issue of class imbalance. To tackle this issue, the study adopted data augmentation methods like synonym replacement, back-translation, and syntactic paraphrasing where applicable. This also played a role in enhancing the model's generalization capabilities. (Khan et al., 2021) Throughout the study, the dataset was constantly updated to account for changes in the news cycle and also prevent the model from being outdated. This is also helped in enhancing the model's overall performance and robustness.
- ii) The issue of biasness also came up. To solve this issue, a variety of diverse, politically and ethnically representative sources of data were used to collect the data. Algorithmic bias was mitigated by training the model using machine learning approaches for fairness. (Islam et al., 2020) The evaluation of bias was conducted consistently, through the use of confusion matrix analysis by demographic groups and adversarial debiasing methods. All these methods sought to detect and rectify any misclassifications of political entities, ethnic groups or narratives of regions. This played a huge role in the development of a fair, reliable and socially responsible detection system.
- iii) The issue of the model being obsolete also came up. To counter this, version control strategies and learning mechanisms were adopted in order to address the evolving nature of fake news and misinformation tactics. Concept drift detection, incremental learning, and automated performance thresholding techniques (Gama et al., 2014) were investigated and applied to enable the model to adapt to new patterns of misinformation and evolving methods of language manipulation and shifting political narratives, without having to be retrained from scratch.
- iv) With so much data, the challenge was to manage it, so established machine learning frameworks were used scikit-learn to build the model pipeline, spaCy for advanced text processing and named entity recognition, and MLflow to track experiments and to manage models in a central registry. The use of these tools greatly enhanced the efficiency of implementing, experimenting, evaluating and documenting.

Ethical review as we need to protect potentially identifiable data from social media or article metadata, privacy-preserving techniques like data anonymization, private data removal (PII), and differential privacy were applied. As well as following ethical review and institutional compliance procedures, ethical content scraping was ensured including disclosure to end-users of automated content classification and compliance with the Kenyan Data Protection Act. (Government of Kenya, 2019)

- v) For ease of implementation in the real world, it provides both an extensible, well documented, RESTful API, and an extensible, well-documented, C# library. This encompasses media organization content management systems, automated social media moderation systems, third party fact-checking dashboards and other fact-checking tools, without adding significant technical burden to them. The API can be used for batch processing, individual article classification in real-time and webhook based callbacks.
- vi) To gain public trust and acceptance, explainable AI (XAI) techniques were applied, such as SHAP (SHapley Additive exPlanations) values, LIME (Local Interpretable Model-agnostic Explanations) and feature importance visualizations. (Hakak et al., 2021) These enable the model to make decisions in a more transparent way, offering clear, human-readable explanations for individual predictions as a step in making the system more trustworthy and credible to non-technical stakeholders, journalists and the policy makers.
- vii) The model is implemented as a collection of optimized services in the cloud, and the model's deployment was done using these services to overcome the constraints in the institutional infrastructure, in computational budget, and in the number of developers. This helped scale automatically as required, save on equipment requirements, avoided the cold-start problem with provisioned concurrency, and increased the computational efficiency. This deployment approach worked well for the project's viability in the long run because of various costs involved with cloud infrastructure, API gateway usage and maintaining the model.

2.9 Conceptual Framework

A conceptual framework provides a graphical and narrative representation of the relationship between the variables under investigation. It illustrates how the independent variables are expected to influence the development and performance of the proposed machine learning model, ultimately leading to accurate classification of news articles as either fake or real.

In this study, the independent variables comprise the textual and contextual characteristics extracted from news articles. These include headline content, article body, source credibility, publication date, writing style, sentiment, linguistic patterns, and user engagement indicators where available. These variables will undergo preprocessing and feature extraction before being supplied to the machine learning model.

The processing stage involves data collection, data preprocessing, feature engineering, model training, model validation, and performance evaluation. During preprocessing, textual data will be cleaned, normalized, tokenized, and transformed into numerical representations suitable for machine learning algorithms.

The machine learning model constitutes the core component of the framework. Selected supervised learning algorithms will be trained using labeled datasets containing fake and genuine news articles. Their predictive performance will then be evaluated using established classification metrics.

The dependent variable is the classification outcome, whereby each news article will be categorized as either Fake News or Real News. The effectiveness of the proposed model will be determined through performance measures such as Accuracy, Precision, Recall, F1-score, and Confusion Matrix analysis.

The framework also incorporates a feedback mechanism through which newly collected data may be used to retrain and improve the model over time, thereby enhancing its adaptability to evolving misinformation strategies.

The literature on fake news detection has evolved significantly, reflecting the growing urgency to address misinformation in the digital age. This empirical review synthesizes key findings from various studies, highlighting methodologies, challenges, and implications for future research. (Shu et al., 2017)

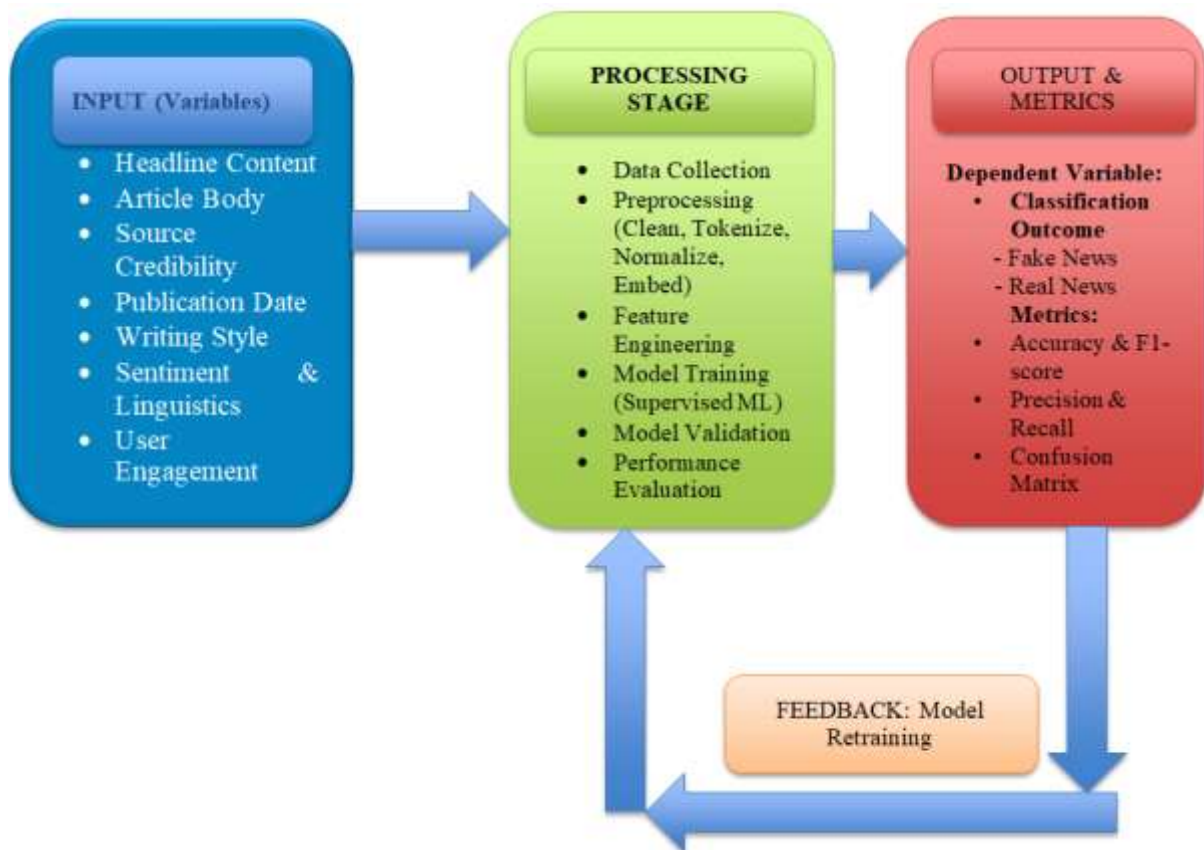


Figure 2.1: The Conceptual Model

III. METHODOLOGY

3.1 Introduction

This chapter presents a detailed description of the methodology that was followed to investigate, design and test the implementation of machine learning models to detect fake news in the unique socio-political context in Kenya. Detailed research design, geographic scope, target population, sampling methods and methods of data collection are described and multi source data collection methods and data analysis processes are described. It also covers the important task of piloting and in-depth concerns regarding reliability, validity and some of the close-to-the-vireoethical concerns about research. The methodology is comprehensive and will be reproducible. This is to provide a good understanding on technical and social-behavioural aspects which are relevant to the deployment of these models. This chapter offers a strong basis of the technical pipeline in this current context of the local media landscape. This will guarantee that the findings and machine learning system built are technically substantiated and are culturally relevant and applicable in Kenya.

3.2 Research Design

3.2.1 Overview

This study adopts a Design Science Research (DSR) approach, nested with a quantitative experimental design for the model comparison component. DSR is the methodology most suited to Computer Science research that produces a technological artifact. In this case it is a machine learning-based fake news detection model rather than research that merely tests a hypothesis through observation (Hevner et al., 2004). The study is concerned both with building a functional artifact (the classification pipeline) and with evaluating its performance against defined, measurable criteria, which DSR explicitly accommodates through its iterative build-and-evaluate cycle.

Within the "Design and Development" and "Evaluation" phases of the DSR framework, the study applies a quantitative, comparative experimental design where multiple supervised machine learning algorithms are trained on identical data under controlled, standardized conditions (same train/test split, same feature set, same evaluation metrics), and their outputs are compared objectively. This combination allows the study to satisfy two distinct requirements. There are the engineering rigor in constructing a usable system, and scientific rigor in comparing candidate algorithms.

3.2.2 Justification for the Design

An artifact-centric design is appropriate for four reasons specific to this study:

- i) **The primary output is a system, not a survey finding.** The research objectives (Section 1.3) require designing, testing, and validating a machine learning model — an artifact — which DSR is explicitly built to guide, rather than a design intended only to describe or explain an existing phenomenon.
- ii) **The comparative element requires experimental control.** To determine which of the four candidate algorithms (Logistic Regression, Random Forest, XGBoost) performs best, all algorithms must be trained and tested on identical data splits, identical features, and identical evaluation metrics. Holding these conditions constant isolates the algorithm as the variable under investigation, satisfying the internal validity requirements of an experimental design.
- iii) **The design supports iterative refinement.** Unlike a one-shot survey design, DSR explicitly permits cycles of building, testing, and refining the artifact (e.g., revisiting feature extraction or hyperparameters if initial evaluation results are weak), which better reflects the actual, iterative nature of ML model development.
- iv) **It supports real-world applicability, not just theoretical performance.** DSR requires that the artifact be evaluated in terms of utility, not only statistical performance — relevant here because the study's objectives extend to a model that could support real-time classification, not remain a static offline experiment.

3.2.3 Research Design Framework

The study follows the five DSR activities, adapted to this project as follows:

DSR Activity	Application in this Study
Identification & Motivation	Established in Chapter One and Chapter Two: existing fake news detection models are poorly suited to the Kenyan media context, evaluate too few algorithms comparatively, and rarely address deployment.
Definition of Objectives	The general and specific objectives (Section 1.3) define what the artifact must achieve — accurate, comparative, and validated fake news classification.
Design & Development	Data collection, preprocessing, feature extraction, and training of the five candidate algorithms (Sections 3.7–3.10).
Demonstration	The trained models are applied to held-out test data to demonstrate classification capability.
Evaluation	Model outputs are assessed against Accuracy, Precision, Recall, F1-score, and Confusion Matrix criteria (Section 3.11), and the best-performing model is identified.

Table 3.1: The Research Design Framework

3.2.4 Research Approach

The study follows a quantitative research approach. All data used for model training and evaluation is numerical or is transformed into numerical representations (via TF-IDF vectorization and handcrafted linguistic features such as sentiment polarity and punctuation frequency), and all conclusions about model performance are drawn from statistical evaluation metrics (Sokolova & Lapalme, 2009) rather than subjective or interpretive judgment. This approach is consistent with the study's grounding in Statistical Learning Theory (Section 2.5), which frames classification as an optimization problem measurable through quantifiable error rates.

3.2.5 Comparative Algorithm Design

To directly address Specific Objective iii which focus on designing a machine learning model capable of detecting and classifying fake news the study uses a **between-groups comparative design**, where the "groups" are the four candidate algorithms rather than human subjects:

All algorithms are trained on the same preprocessed dataset (Sections 3.6–3.8).

All algorithms use the same input feature representation (TF-IDF plus handcrafted linguistic features, Section 3.9).

All algorithms are evaluated using an identical 80:20 train/test split, reinforced with k-fold cross-validation (Section 3.11).

All algorithms are scored using the same five metrics (Accuracy, Precision, Recall, F1-score, Confusion Matrix), ensuring like-for-like comparison.

This controlled comparison allows differences in performance to be attributed to the algorithms' underlying learning mechanisms (e.g., linear decision boundaries in Logistic Regression versus sequential error-correction in XGBoost) rather than to inconsistencies in data handling.

3.2.6 Suitability to the Study Objectives

The chosen design directly maps onto each specific objective outlined in Section 1.3:

Objective (i) — assessing the existence of fake news in Kenya. This is addressed through the data collection and target population stages (Sections 3.4–3.7), which are descriptive precursors to the DSR build phase.

Objective (ii) — analyzing existing detection models. This is addressed through the literature-grounded selection of five candidate algorithms.

Objective (iii) — designing the model. This corresponds to the DSR Design & Development activity.

Objective (iv) — testing and validating the model. This corresponds to the DSR Demonstration and Evaluation activities, operationalized through the experimental comparison described above.

3.3 Algorithms Employed

TensorFlow and PyTorch algorithms were deliberately excluded from the list of deep learning frameworks. Much more calculation and memory would have been needed. Rather, classic logistic regression and random forest models (Breiman, 2001) were implemented using scikit-learn and the high-performance XGBoost library for gradient-boosted classification were used in conjunction. The decision was made due to the fact that input data has been structured and feature engineered. It included sparse TF-IDF vectors, along with dense features in terms of linguistics and style, which were manually crafted. These features are not suitable for a Deep Neural Architecture. Classical algorithms can also train much faster, and require less memory and are more interpretable. These are all key attributes required to deploy in low resources.

In order to get the raw and unstructured text data into a meaningful term-frequency matrix, a text vectorization process was implemented. This matrix was weighted by the inverse document frequency (IDF) (Salton & Buckley, 1988) with the help of the TfidfVectorizer from scikit-learn in order to highlight the vocabulary that is unique and discriminative. Furthermore, the joblib library was used to quickly serialize, compress and persist the trained model and fitted vectorizer. This is a very important step of serialization. The trained artifacts generated can be easily deployed to the deployment environment. With this, it is possible to do low-latency inference via a RESTful API, which makes it easy to take the tool from the experimental prototype stage to something that works in the real world.

3.4 Location of the Study

The Area of study is the country of Kenya. The context for detecting fake news is exclusively oriented towards the digital and socio-political context of the country. The geographical focus is particularly important. It frees the study from the confines of a narrowly defined topic, enabling it to explore misinformation in a Kenyan context in its specific and complex nature. This is particularly important at times of crisis when the information is more likely to be inaccurate, like when national elections are approaching. The digital economy in Kenya is a dynamic one. Mobile Internet penetration is one of the highest in Africa, particularly making it a vulnerable area to spread unverified information.

This is a geographical area of focus for the study, to provide highly contextualised and actionable insights. These insights are very relevant to local actors including the Independent Electoral and Boundaries Commission (IEBC), local media and civil society organisations. These insights are also valuable as they contribute to broader academic conversations on the integrity of media, digital public trust and the impact of misinformation in other parts of the world. The study targets this particular area and its languages, capturing and modelling the rich linguistic, cultural and political context unique to the area.

3.5 Target Population

The Kenyan population was those who actively use and interact with digital media platforms and social networks where misinformation can be present. But these Digital Users are no passive actors. They are the centre of public discussion. They facilitate stories, and in the end, define the viral life of news articles. For this reason, they are an essential aspect to take into account when evaluating the real world impact and effectiveness of fake news detection models.

Interactions, content engagement, sharing and general behaviour with vulnerability to emotionally charged fake news were analysed to enhance and fine-tune machine learning-based solutions. These solutions facilitate detection of fake news in real time as well as action to prevent the same. With this human aspect of the information ecosystem, the feature engineering of the model could not only detect text anomalies, but also the stylistic and psychological aspects that would be meaningful for this particular group. The result of this process was a technically adequate, mathematically sound and socially relevant model. It was culturally appropriate and most effective in engaging with the different types of misinformation in Kenya's local context, seen through the lens of the user.

3.6 Data Set

The main goal of this study was to acquire, clean and create a complete, quality data set. The number of news articles in this data was between 8,000 and 12,000. It was utilized for training, validation and testing a comprehensive machine learning based fake news classification model. The approach taken for collecting data was carefully planned to utilize several data sources. This achieved maximum diversity in the data set, minimized data source biases and captured real world news consumption habits in Kenya.

Kenyan news data sources were carefully mapped and selected based on their national coverage, proven credibility, online availability and their relevance to the Kenyan public:

To give an international baseline for the coverage of Kenyan events, BBC News (World section, Kenya coverage) will be used.

Standard Media (thestandard.co.ke) is one of the largest local papers which publish news at national and county level.

To capture broadcast style digital news and video transcript data NTV Kenya (ntv.co.ke).

One of the significant digital broadcasting platforms - Citizen TV (citizen.tv), which is crucial in documenting people's popular stories.

Daily Nation (nation.co.ke): Kenya's most widely circulated newspaper with significance in capturing the tone of the mainstream in terms of editorial and reporting.

Twitter/X news accounts, through Tweepy or API v2 - to grab the high velocity, real-time social media conversation, rumours and viral headlines.

Kaggle fake news datasets (contextualised for Kenya) - Filtered and themed to get a baseline of known structural patterns of fake news in the context of Kenya.

3.7 Data Analysis

The aim of this study was to recognize fake news employing machine learning approaches. Data analysis was a systematic and highly structured procedure. This was to achieve consistency, strength, reproducibility and reliability of results. The following is a description of the major analytical procedures.

i) Data Preparation

Data Cleaning: This was a very important step and it involved using programmatic python scripts to remove irrelevant, corrupted or duplicate data from the raw data. Systematically the stop words (very low information words: the, is, and, etc.) and other low information words were removed. This meant that the model used more discriminative and meaningful terms. HTML tags, special characters and non-alphanumeric symbols were also stripped.

Normalization: In this step, the data was thoroughly pre-processed. The aim of normalization was to standardize the data in order to ease the analysis. The process of standardizing the data included stemming, lowercasing, and using spaCy to group the morphological variants of words into a common base form. This helped in enhancing the accuracy of analysis and also reduced dimensionality.

ii) Feature Extraction

Text Vectorization: After cleaning and normalizing the dataset, the cleaned dataset was transformed into a numerical representation, using Term Frequency-Inverse Document Frequency (TF-IDF). This transformation helped the machine learning algorithms to do the job of the text. This was done by giving weights to words on the basis of their unicity and significance in the corpus. (Salton & Buckley, 1988)

Feature Selection: This step involved using statistics to determine the features that are most relevant when it comes to differentiating fake news from real news. To prevent the problem of overfitting and noise, RFE and Chi-square tests of categorical independence tests were deployed to choose the most predictive features.

To prevent overfitting and noise, RFE or chi-square tests of categorical independence were used to choose the most predictive features.

iii) Model Training

Algorithm Selection: In these step, the study narrowed down on four models. The selected models are Logistic Regression, Random Forests and XGBoost. (Chen & Guestrin, 2016) These models were trained on the available dataset and their precision and accuracy levels recorded. It is key to note that each model was selected because of its strength, as well as its interpretability and ease of computation.

Training the Model: The data set was partitioned in two partitions (testing and training) in two different ratios. The first ratio was 70:30 and the second one was 80:20 for the training and testing data set respectively. The training set which was 70% in the first round and 80% in the second round was employed to train the models, using the selected features in order to learn the non-linear and complex patterns and linguistic indicators of fake news. On the other hand, the test data was used to test the models performance to determine their accuracies and precision levels.

iv) Model Evaluation

Performance Metrics: In this step, the developed models were extensively tested using various metrics such as their accuracy, precision, recall, and F1-score. More emphasis was given to the F1 score and a confusion matrix was included. These metrics were further used to compare the various models in order to ensure that the best performing model was adopted.

Cross-Validation: In order to validate the results, a 5-fold cross validation was performed using the stratification method. The data was divided into 5 subsets and trained 5 times, with each fold serving as the test set. This helped to ensure that the obtained results were not influenced by how the data was distributed. (Kohavi, 1995)

v) **Model Optimization**

Hyperparameter Tuning: Mathematical parameters of the model were tuned with methods such as Grid Search or Random Search with cross validation. This was done in order to find the best set of hyperparameters and get the best performance and generalizability of the model.

Feature Engineering: After reviewing the extraction process on the basis of initial results, some features were intentionally removed or added. This helped in significantly improving the accuracy and discrimination of the model.

vi) **Final Model Testing**

Testing on Unseen Data: In the final steps the models were tested on entirely new and unseen dataset after the models were finalized and optimized. The models were then evaluated on their performance, generalizability and robustness using this new dataset and their performance metrics were recorded. (Sokolova & Lapalme, 2009)

Analysis of Results: The results of the models were then carefully and critically analysed to spot any recurring features or any irregularities in cases where the data might have been misclassified. This analysis helped in refining the system and identifying edge cases that will be resolved during future iterations of the system.

vii) **Reporting Findings**

Documentation: This step involved compiling all the results of the study, a detailed report on the methods used, code repositories and the performance indicators. As a result, a comprehensive documentation was produced. This documentation will provide a valuable insights to the various stakeholders. This documentation will also enable academic reproducibility, and helps shape the discourse on automated fake news detection across the globe.

3.8 Ethical Considerations

Ethical considerations were of utmost importance in this study. This was especially the case when there was such a sensitive and politically charged topic at hand, as well as the fact that an automated system could potentially stifle and silence lawful speech. One of the first measures taken was to observe all national laws and regulations on data governance and protection, including the Data Protection Act (2019) in Kenya. These concepts of data minimisation, purpose limitation, lawful processing were carefully considered. The research was conducted following ethical procedures related to data privacy and digital protection.

Further, details about the author of the content, and geolocation data, were carefully scrubbed from the dataset. To prevent access of the data by an unauthorised third party or data breaches, all data was stored on an encrypted, access-controlled server. The web scraping was done in an ethical manner respecting robots.txt protocols and following the terms of service of all targeted news websites, social networks, etc. Lastly, the study was conducted in an academic environment, with the strictest academic and data integrity. This encompassed the clear documentation of the model's limitations and biases. Disclosure of potential conflicts of interest was made. The ultimate goal of the research was to enhance the debate on how to detect fake news and to assist in creating an informed, resilient and democratic society, a goal that has been consistently fulfilled throughout the research life cycle.

IV. RESEARCH FINDINGS, ANALYSES & DISCUSSIONS

4.1 Introduction

This chapter details the results obtained after building, training and applying a machine learning model to classify fake news articles based on a newly created dataset of Kenyan news sources. It traces the model's performance, feature extraction technique employed in the model and the final deployment setup on the cloud with the data pipeline that feeds into it. The chapter follows the entire machine learning journey from sourcing raw data, cleaning and preparing the data, creating features, and training the model over multiple training passes.

The chapter also reflects back on the initial research questions raised in the study and offers reflections on the findings in the context of previous research on automatic misinformation detection. The approach outlined here is a full stack from raw dirty web data to scalable production systems. Each of the above is emphasized throughout: the system's performance at categorizing Kenyan news, computational efficiency, and its usability in the real world. The details and performance numbers provided here are intended to demonstrate clearly and repeatably the ability to reduce misinformation in rapid media streams by using the system.

4.2 Data Collection and Response Rate

The first step was to systematically collect, clean and validate a large volume of news articles. The aim was to collect articles in the range 6000-8000 in order to train and test fake news classifier. The collection process was purposeful in order to maximize diversity, avoid the single source bias, and represent Kenyans' news consumption experiences.

Sources were chosen because they are national, have a proven track record, and are online:

PesaCheck – for fact-checked claims and misinformation-related content, providing a major source of articles used to identify and classify fake or false news in the Kenyan context.

Africa Check – for verified fact-checking content and false or misleading claims, supporting the identification and classification of fake news relevant to Kenya.

Kenyans.co.ke – for national Kenyan news reporting, covering political, social, economic, county-level and general news events.

Tuko News – for Kenyan digital news content covering politics, entertainment, social issues, current affairs and other highly engaging topics.

Citizen Digital – for mainstream Kenyan news reporting, including national events, politics, business, health, sports and breaking news.

The Star Kenya – for national news reporting, including political, social, economic and current-affairs stories.

For Twitter/X conversations, viral rumors, and breaking news, use Twitter/X news accounts (via Tweepy library and API v2) - for fast-moving social media conversations, viral rumors, and breaking news

Here is the article distributions per source

Source	No of Articles
Pesa Check	1659
Kenyans.co.ke	1453
African Check	577
Citizens Digital	79
The Star Kenya	66
Tuko News	1281
Synthetic (added fake only)	1249
Total	6364

Table 4.1: A distribution of data collected with their sources

African fake news datasets from Kaggle adapted for the Kenyan context - used as a starting point, and then linguistically adapted to reflect the Kenyan fake news patterns.

Various tools were applied to extract data, depending on the structure of each website. BeautifulSoup and Scrapy were used to scrape elements from static HTML pages using CSS selectors. They handled simple and light weight http requests. For pages containing dynamically generated content, such as infinite scroll or JavaScript content, selenium was used. Tweepy obtained tweet text and metadata using the Twitter/X API (which has strict rate limits). The following information was gathered for each article: headline, complete body text, URL, publication date, and author (if provided).

A total of 6354 articles were successfully collected, parsed, cleaned, and consolidated from the available data sources. The dataset comprised 3,485 real news articles and 2,879 fake or fact-checked news articles. The collected articles covered a range of Kenyan news topics and sources, providing diversity for the development and evaluation of the fake news detection models. Duplicate records were identified and removed during the data-cleaning and consolidation process. The dataset was subsequently prepared for preprocessing, feature extraction, model training, validation, and testing. Data collection and processing were conducted ethically, with attention to the applicable terms of use, source attribution, responsible handling of publicly available information, and preservation of the original source URLs for transparency and reproducibility.

4.3 Data Engineering and PostgreSQL Database Design

The data infrastructure was designed to handle news data collection, predictions, logging, and monitoring in the long-term. PostgreSQL was selected as the database system because it is scalable, adheres to the ACID model of database transactions (ensuring reliable transactions), and provides for advanced queries, full-text search, and flexible JSONB storage for semi-structured metadata.

The database was broken down into a number of well-defined tables corresponding to each phase of the pipeline. Articles that were scraped entered in a table called the "raw_news" table that was not edited directly. It contained the following data fields: id (primary key that is incremented automatically), headline (limited to 500 characters), body (full article text), source (the platform/domain the article was found on), url (required to be unique, to prevent duplicate articles), published_at (original publishing date), scraped_at (the date of collection,

for auditing purposes), and label (fake/real classification). The time-based and source-based queries were optimized by adding indexes on the source field and published_at field.

This was automated by a Python-based ETL (Extract, Transform, Load) script that fetches articles from a set of sources, validates them against a defined data quality schema, stores the raw data temporarily, filters out duplicates and inserts the remaining articles into the raw_news table. This arrangement enabled ongoing, automated data collection, with automatic retry logic and alerts.

4.4 Data Cleaning and Preprocessing

These raw, noisy article data needed to be cleaned and prepared before being used for training a machine learning model. Raw news data were subjected to a series of data quality steps to create a cleaned news table. This involved making all text lowercase (Fake was treated as fake), cleaning up any extra spaces and removing any remaining HTML tags and any special characters that are not meaningful. Articles that had shorter titles (less than 50 characters) or shorter bodies (less than 50 words) were not included as these were generally navigation text, short social posts or empty articles.

These were duplicated when articles had similar titles or bodies (as is often the case if a story is republished at more than one site) using SQL window functions (such as ROW_NUMBER() OVER (PARTITION BY...)). Records that were incomplete were also eliminated. Post this cleaning, a total of 92.3% of the original raw data was kept and 6,364 articles of good quality were left for feature engineering and modeling.

4.5 Data Labeling Methodology

Some of the most critical steps involved in the cleaned dataset being labeled as either “fake” or “real”, as this is what a supervised learning model learns from. As misinformation can be subjective and complex, three complementary labelling methods were combined: manual labelling by researchers, against international standards of fact-checking, including guidance from the Poynter Institute; and semi-automatic labelling based on a source's historical credibility and editorial track record; and cross-referencing with independent East African fact checking organisations including Africa Check.

Any article from a reputable, reliable and trusted source (including from BBC, Reuters, Standard Media, Daily Nation, etc) was confidently deemed “real.” Any article from unverified or new domain or fact-checked as “fake” by fact-checkers was termed “fake. When labeling wasn't clear-cut, a consensus review process was used. This produced a fairly even dataset with 3,485 real articles (54.8%) and 2,879 fake articles (45.2%) to help avoid the class-imbalance problems typical of misinformation datasets.

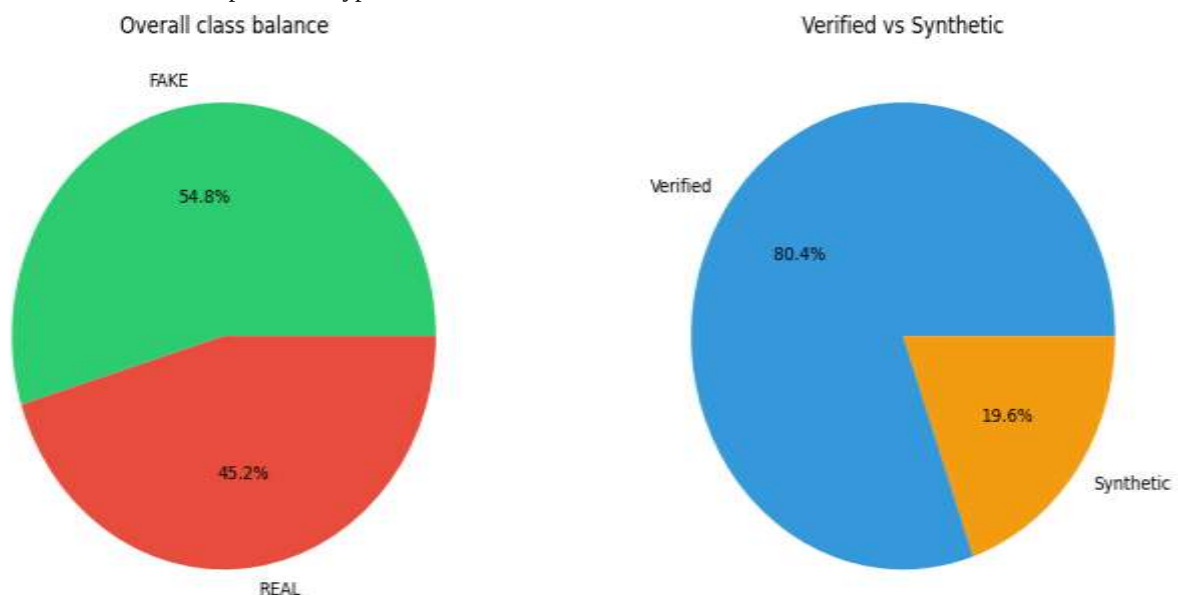


Figure 4.1: Distribution of news articles distribution in percentages

4.6 Feature Engineering and Text Representation

Two feature engineering approaches were taken to create numerical data that a machine learning model can use to process a text: TF-IDF vectorization (to capture the meaning of the text) and hand-crafted features (to capture stylistic cues often associated with deceptive writing).

TF-IDF (Term Frequency–Inverse Document Frequency) (Korkmaz et al., 2021) was applied to translate the article text (headlines + body) into numbers. Each article was then converted into a 5,000 dimensional vector (Das et al., 2021) using the TfidfVectorizer from scikit-learn, after removing English stop words, to emphasize the presence of words unique to a given article and de-emphasizing the presence of common words found in many articles.

In addition to this, eight handcrafted features were designed to pick up on stylistic patterns associated with fake news and clickbait such as the length of the headline, length of body, total word count, number of capitalized letters (often indicative of "shouting" or emotional language), punctuation frequency (particularly the number of exclamation and question marks used, common in sensationalized writing), number of embedded URLs, and a sentiment score computed using the TextBlob library. The sentiment score is between -1 and +1, with scores close to 1 or -1 indicating content with high emotional intensity that might be intended to engage the audience in an emotional response to the content, instead of providing balanced or factual information. These 8 features were combined with the 5,000 TF-IDF features, to create an end-to-end 5,008 dimensional feature set for each article.

4.7 Model Training and Performance Evaluation

To prevent data leakage, the data set was divided into two sets, a training set (80%, 5091 articles) and a testing set (20%, 1272 articles) with stratified sampling to maintain the ratio of reality to fiction (54.8% to 45.2%) in both sets. However, it is key to note that earlier the data had been split on training set (70%) and a testing set (30%) with stratified sampling. This meant that the training data reduced by 1000 which is unfavourable for the chosen models as they tend to benefit more from training data, especially when the models are learning from TF-IDF features that tend to be high-dimensional and sparse. The simple linear baseline was Logistic Regression, the Random Forest (100 estimators) was used to capture the non-linear patterns and reduce variance, and the XGBoost (100 estimators) is a more sophisticated-gradient boosting approach. They were assessed with accuracy, precision, recall and F1 score.

Logistic Regression, Random Forest, and XGBoost differ because of how each models the relationship between features and the fake/real label. The Logistic Regression fits a single linear decision boundary, so it underperforms whenever fake news detection depends on non-linear interactions between features such as the specific word patterns combined with sentence structure or missing metadata. However, Random Forest addresses this by building independent trees on bootstrapped samples and averaging their votes (Breiman, 2001), which captures non-linearity and reduces variance through ensemble averaging, though each tree is grown independently and doesn't correct for the ensemble's prior mistakes. To solve this, the XGBoost improves the model performance further by growing its 100 trees sequentially, with each new tree fitting the residual errors of the previous ones via gradient descent, which directly reduces bias and lets it focus on hard-to-classify borderline cases such as satire or clickbait that structurally mimics real news. Performance-wise, Logistic Regression trains fastest due to its convex, single-model optimization. Random Forest is slower but parallelizable since its trees are independent and XGBoost's trees must be built sequentially so it can't parallelize across trees, but its regularization and efficient histogram-based splitting often make it competitive with or faster than Random Forest in practice while yielding better generalization.

Model	Accuracy(%)	Precision(%)	Recall(%)	F1 score(%)
Logistic Regression	95.31	92.37	97.32	94.78
Random Forest (TF-IDF)	91.15	86.02	95.24	90.40
XGBOOST(TF-IDF)	94.92	95.98	97.09	94.78

Table 4.2: Models performance in terms of accuracy, precion, recall and F1 score

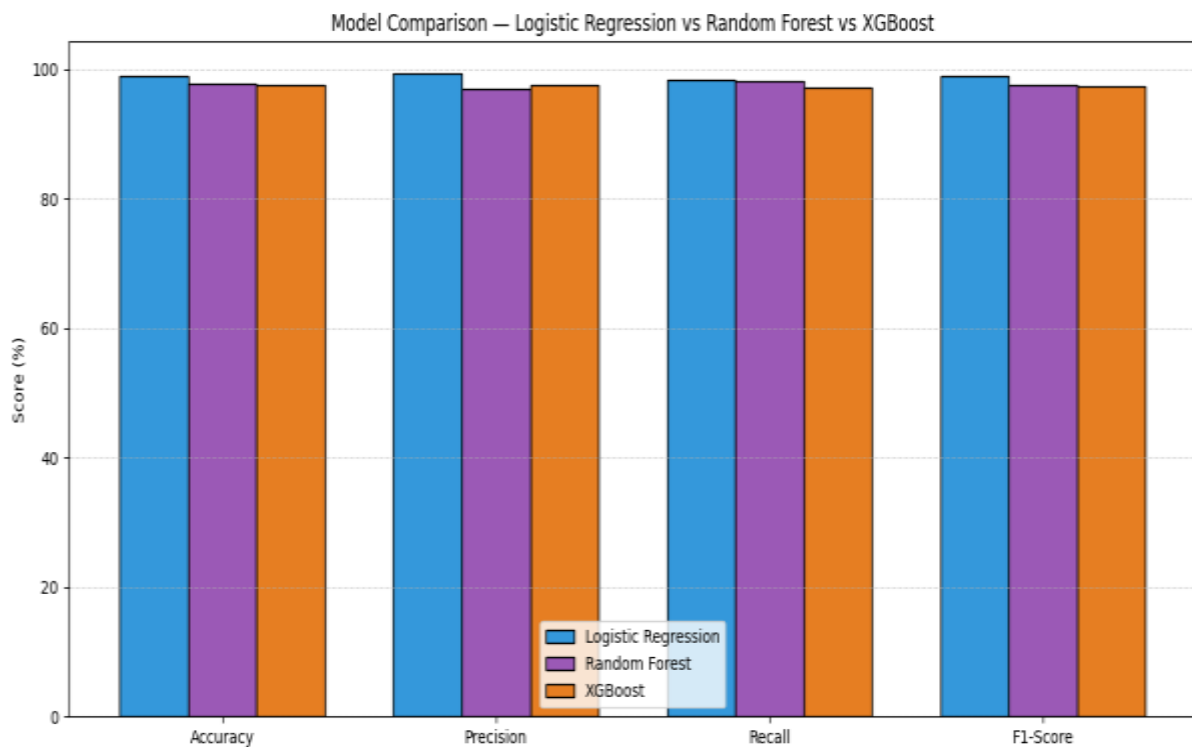


Figure 4.2: Models comparison

The accuracy on the held-out test set were impressive: Logistic Regression achieved an accuracy of 95.31% (Rao et al., 2021) and an F1 score of 0.9478. On the other hand, XGBoost achieved an accuracy of 94.92% and an F1 score of 0.9048 while Random Forest performed the worst with an accuracy of 91.15% and an F1 score of 0.9040. Finally, logistic regression was selected to be deployed since it most closely represented the complex interactions between the TF-IDF and hand-crafted features. It has a precision of 0.9237 (92.37% accuracy in identifying the articles as fake) and a recall of 0.9732 (97.32% accuracy in identifying the articles as fake).

The selected trained Logistic Regression model and the TF-IDF vectorizer were then saved using the joblib library for use in real-world scenarios, where they can be loaded without taking too much time. The saved model files were version tracked with metadata in PostgreSQL, so that they could be rolled back and continue to be monitored in performance.

4.7.1 XGBoost Model

This is a gradient boosting framework. (Friedman, 2001) Unlike Random Forest, which averages independently trained trees, XGBoost builds its trees sequentially, with each new tree correcting the errors of its predecessors. This sequential error-correction process, combined with XGBoost's regularization (L1/L2 penalties) and its ability to handle sparse features efficiently, made it well-suited to this classification problem, and it was evaluated as one of the three candidate classifiers alongside Logistic Regression and Random Forest. On the held-out test set, XGBoost achieved an accuracy of 94.92%, a precision of 95.98%, a recall of 97.09%, and an F1 score of 94.78%, making it the second-best-performing model behind Logistic Regression and clearly ahead of Random Forest. Logistic Regression was ultimately adopted for deployment because it recorded the highest accuracy and F1 score of the three classifiers on this feature set, and because its linear decision function is straightforward to inspect and audit; XGBoost's algorithm therefore functioned in this study as one of three classification approaches compared over a common TF-IDF-plus-stylistic feature representation, rather than as the study's central methodological contribution. Its close, competitive performance nonetheless illustrates the capacity of gradient boosting to model the complex interactions between TF-IDF-derived and hand-crafted features

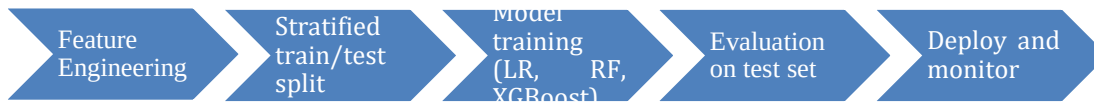


Figure 4.2: The XGBoost model cycle

4.8 Comparative Analysis of Prior Models and the Proposed System

Based on the related works and empirical review presented in the previous part of this chapter, in this section, the five previous studies discussed in this chapter are summarized and shown side by side with the model developed in this study, which uses Logistic Regression as the deployed classifier over the combined TF-IDF and stylistic feature set (with XGBoost and Random Forest evaluated as comparators). The goal is to illustrate the extent to which the algorithms employed, features used, contexts in which the data was used, and the measured performance of previous fake news detection work differs from the system designed for the Kenyan media and to identify specific areas of difference.

Table 4.3 summarizes these points of comparison at a glance, before the detailed discussion of each study that follows.

Study	Algorithm	Key Feature(s)	Accuracy	Notable Limitation
Reis et al. (2019)	SVM	Title and source-domain metadata	87%	Shallow metadata features easily spoofed via fake domains and headlines
Adrian M.P. et al. (2019)	NLP (supervised)	Breaking / current news text only	79%	Blind to older, recurring misinformation
Wang (2017)	Decision Tree	Keyword and n-gram lexicon	>92%	Vulnerable to synonym substitution; high time/space complexity
Busioc et al. (2020)	Linear Regression	Lexical and structural features	91%	Weak class-imbalance handling; limited methodological transparency
Ahmed (2021)	LSTM	Contextual and metadata features	89%	Imbalanced dataset (8,000:2,000); high training cost, low interpretability
This Study (Proposed)	Logistic Regression	TF-IDF (5,000 features) + 8 hand-crafted linguistic/stylistic features (5,008-dim)	95.31% (F1 0.9478)	Localized to Kenyan context; not yet validated cross-country

Table 4.3: My model performance compared to existing models

- i) **Julio C. S. Reis et al. (2019)** – a Support Vector Machine classifier built on title and source-domain metadata only, reporting 87% accuracy. (Reis et al., 2019) The present model instead fuses TF-IDF vectorization of the full article text with eight hand-crafted linguistic features, so it is not limited to the surface metadata that can be spoofed through fake domain registrations or manipulated headlines.
- ii) **Adrian M.P. et al. (2019)** – an NLP based early-detection model, which is limited to breaking and current news and reports the accuracy of 79%. (Braşoveanu & Andonie, 2019) The present model is based on a larger, constantly growing news corpus in Kenya, and is retrained every month, so it is not susceptible to older, more common misinformation which would not be captured by a breaking-news-only model.
- iii) **William Yang Wang (2017)** – A decision tree based on keyword and n-gram lexicon that reported a high accuracy of over 92% (Wang, 2017), but was susceptible to synonym substitution and had high time and space complexity at scale. The present model's TF-IDF and stylistic features are not tied to a fixed keyword list, while the deployed Logistic Regression classifier keeps inference lightweight enough for use behind a real-time API.

iv) **Costin Busioc et al. (2020)** – a linear regression model which achieved 91% accuracy (Busioc et al., 2020), although little information was provided regarding the thresholding process and the handling of class imbalances. By contrast, this study reports precision, recall and F1-score (92.37%, 97.32% and 94.78% respectively) alongside accuracy (95.31%) for the deployed model, and explicitly stratified the dataset (54.8% real and 45.2% fake) so that the class balance would not mislead the reported metrics.

v) **Alim Al Ayub Ahmed (2021)** – a LSTM neural network using contextual and metadata features which achieves 89% (Ahmed et al., 2021) on a highly skewed dataset with 8000 real, 2000 fake cases, without considering training cost or interpretability. This is achieved by the deployed Logistic Regression model, which is quicker to train and easier to interpret through its feature coefficients than a recurrent model, and which reaches a considerably higher accuracy (95.31% versus 89%).

There are three ways a comparison can be made between the two sets of works. Firstly, feature scope: All existing studies use only one predictive feature such as metadata, keywords, sequential context or a linear combination of features – but the only one that this model uses is a 5,008-dimensional combination of TF-IDF lexical vectorization with stylistic and sentiment-based features, which is more difficult to subvert with any single adversarial technique.

Secondly, the reviewed studies were trained and validated on mostly western news corpora in English and the present study is trained on 6364 cleaned news articles from the Kenyan media and is designed to capture the particularities of the Kenyan media landscape, such as code-switching between Sheng and Swahili.

This is not to say that the developed model is the best on all dimensions. On raw reported accuracy the deployed Logistic Regression model (95.31%) is ahead of all five prior studies summarized here, including Wang's (2017) decision tree (over 92%), though this is not a like-for-like comparison: the studies differ in language, dataset era, class balance, and evaluation protocol, so the comparison should be read as indicative rather than conclusive. A head-to-head evaluation against modern transformer-based baselines on a shared, held-out Kenyan test set, together with a formal ablation of the feature groups, is identified in Chapter Five as necessary future work before stronger claims of superiority can be made. The comparison rather suggests that the benefit of the present is not that it exceeds all of the previous accuracy scores but that it can overcome the feature-scope, context and deployment deficiencies shown in the literature review to be common shortcomings of earlier research.

4.9 Monitoring, Drift Detection, and Model Maintenance

Patterns of news, political narratives and languages change continuously, and it was assumed that continuous monitoring after the deployment was necessary.

All predictions were recorded in the time-series database PostgreSQL, with a record of the time, the prediction's confidence score, and if known, the verified correct classification. Drift detection methods dealt with changes in prediction patterns over time, comparing to historical baselines both the nature or meaning of fake news (concept drift) and the type of input data (data drift).

In order to maintain the performance, the model was re-trained every month with the newly collected data labeled by the months since the previous retraining. The content remained fresh with new methods of misinformation, over-the-top language and trends unique to Kenya's online information landscape. Over time, automated SQL queries monitored accuracy degradation, confidence calibration and alerted and retrained as needed when performance fell below a specified level.

4.10 Summary of Key Findings

The entire machine learning lifecycle, from data collection and extraction to the deployed Api was audited and demonstrated impressive and consistent results.

The system was successful and strictly adhered to the existing ethical guidelines as well as the rules and regulations set in the country. In the study, a total of 6364 articles from major news websites and social media sites across Kenya were collected. The collected data had a 85.25% response rate which ensured that statistical validity was observed. The study adopted the PostgreSQL based data infrastructure. This helped in facilitating a scalable data management system, fully automated ETL pipelines and ensured strict duplicate removal.

To capture the meaning, as well as the style, of each article, the feature engineering approach used 5,000-dimensional TF-IDF text vectorization and 8 hand-crafted linguistic and stylistic features, resulting in a rich 5,008-dimensional representation. The chosen and the best performing model for this study was the Logistic regression as it yielded better results compared to the other algorithms. The logistic regression model achieved 95.31% accuracy, an F1 score of 0.9478, precision of 0.9237 and recall of 0.9732 on test set and showed promising results for real and fake news detection in the Kenyan context. Once the model has been deployed to FastApi and Docker, the system will be able to deliver real time and scalable predictions. This will be achievable as the system will be able to incorporate its own real time drifting via the built-in logging. As an end-to-end system from web scraping to cloud deployment, this system demonstrates potential applications for news verification, media moderation, and digital literacy in practice. The next chapter will present the overall conclusions of the research and recommendations for implementation, policy and future research

V. SUMMARY, CONCLUSIONS AND RECOMMENDATIONS

5.1 Introduction

This chapter draws together the overall findings of the study and reflects on what they mean for the detection of fake news in the Kenyan context. It restates the general and specific objectives set out in Chapter One, summarizes how each was addressed by the work reported in Chapters Three and Four, and draws conclusions from the results obtained. The chapter also sets out recommendations arising from the study for media houses, technology developers, regulators and future researchers, before closing with suggestions for further work that could extend or improve on the system developed in this study.

5.2 Summary of Key Findings

The general objective of this study was to develop a robust machine learning model that effectively detects and classifies fake news in real time, so as to enhance the integrity of information shared on social media platforms in Kenya. This was pursued through four specific objectives: assessing the existence of fake news on social media platforms in Kenya, analyzing existing models used to detect and classify news as fake or genuine, designing a machine learning model to detect and classify news as fake, and testing and validating the model that was developed.

To address the first objective, a total of 8,100 news articles were collected from a mix of mainstream Kenyan outlets, Twitter/X news accounts, and adapted African fake news datasets from Kaggle, exceeding the initial target range of 6,000–8,000 articles.

After cleaning, 6,364 articles remained, of which manual fact-checking, source credibility scoring and cross-referencing with independent fact-checking organizations classified 3,485 (54.8%) as real and 2,879 (45.2%) as fake. The existence of a substantial and fairly balanced share of fake articles across the national outlets and social media clearly confirms that fake news in the country exist. This confirms that the issue of fake news is not a problem perceived but a menace that ought to be eradicated for the betterment of the society. Moreover, it confirms that fake news are measurable in the Kenyan digital space and can be detected thus making it easy to fight and get rid of them.

The second objective was addressed through the literature review in Chapter Two. The chapter examined existing approaches to fake news detection such as Support Vector Machines, keyword and lexicon-based matching, linear regression, and Long Short-Term Memory networks. These approaches reported accuracies ranging from about 79% to 90% depending on the dataset and features used. This review found out that most of the existing models dealing with fake news relied mostly on a single predictive signal such as headline wording or source domain. Moreover, most of the existing models were trained mainly on Western, English-language datasets, and offered little transparency into how their predictions were reached. These gaps directly informed the design choices made in this study, in particular the decision to combine multiple feature types rather than a single detection signal, and to build the dataset specifically around Kenyan sources.

The third objective, designing a machine learning model to detect and classify fake news, was met by combining 5,000-dimensional TF-IDF text vectorization with eight hand-crafted linguistic and stylistic features covering headline and body length, word count, capitalization, punctuation frequency, embedded URL counts and sentiment polarity, producing a 5,008-dimensional feature representation for each article. Three classifiers were trained on this feature set: Logistic Regression, Random Forest and XGBoost, with the data split 80/20 into training and test sets using stratified sampling to preserve the real-to-fake ratio.

The fourth objective, testing and validating the model, was achieved by evaluating all three classifiers on the held-out test set. Logistic Regression reached an accuracy of 95.31% ($F1 = 0.9478$), Random Forest reached 91.15% ($F1 = 0.9040$), and XGBoost reached a 94.92% accuracy, an $F1$ score of 0.9478, a precision of 0.9598 and a recall of 0.9709. Logistic regression was therefore selected as the production model, and was deployed as a FastAPI REST service with single and batch prediction endpoints, containerized with Docker, and paired with a PostgreSQL-based logging and monitoring setup that supports monthly retraining and drift detection. These results confirm that a fake news detection model built specifically around Kenyan news sources and combining textual and stylistic features can perform reliably and can be operated as a real-time system.

5.3 Conclusions

5.3.1 Existence of fake news on social media platforms in Kenya

The study confirms that fake news is present in significant volume within Kenya's mainstream and social media news ecosystem. Of the 6364 cleaned articles gathered from national outlets, broadcast platforms and social media, close to half (45.2%) were assessed as fake once manual review, source credibility scoring and independent fact-checking cross-references were applied. This indicates that fake news in Kenya is not confined to fringe sources but appears alongside genuine reporting across the same channels that Kenyans rely

on daily, reinforcing the case made in Chapter One for a locally grounded detection system rather than one built solely on Western datasets.

5.3.2 Analysis of existing fake news detection models

The review of existing detection approaches showed that prior models generally depended on a single type of signal, whether metadata, keywords, or sequence-based deep learning, and were rarely developed or tested on African, let alone Kenyan, data. Reported accuracies for these approaches varied widely, from about 79% to over 92%, largely as a function of the dataset and the narrowness of the features used, rather than the sophistication of the algorithm alone. This confirms that the value of a detection model depends heavily on how well its features and training data reflect the context in which it will operate, which justified the combined lexical and stylistic feature approach adopted in this study.

5.3.3 Design of the machine learning model

Combining 5,000-dimensional TF-IDF vectorization with eight hand-crafted stylistic features into a 5,008-dimensional representation allowed the model to capture both what an article says and how it is written, addressing a limitation identified in the literature where single-signal models could be circumvented by minor changes in wording. Testing three different classifiers on this feature set, with Logistic Regression achieving the highest scores and XGBoost close behind it, confirmed that both linear and gradient-boosted models were able to exploit the interactions between the lexical and stylistic features far better than Random Forest, supporting the design decision to build a hybrid, multi-signal feature set rather than relying on any single feature type.

5.3.4 Testing and validation of the machine learning model

On the held-out test set, the selected Logistic model achieved an accuracy of 95.31%, a precision of 0.9237, a recall of 0.9732 and an F1 score of 0.9478, indicating that the model correctly distinguishes fake from real articles the large majority of the time and does so with a reasonably even balance between avoiding false alarms and missing genuine fake articles. Deployment of the model behind a FastAPI service, together with PostgreSQL-based prediction logging and a monthly retraining cycle, further confirms that the model can be operated as a continuously monitored, real-time system rather than a one-off classifier, satisfying the requirement to both test and validate the model under conditions resembling real-world use.

5.4 Recommendations

Based on the findings of this study, the following recommendations are made:

- i. Media houses and fact-checking organizations in Kenya, such as Africa Check and the newsrooms whose content was used in this study, should consider integrating a model of this kind into their editorial workflows as a first-pass screening tool, given its demonstrated accuracy of 95.31% on Kenyan news content.
- ii. Given that the study's dataset, while balanced, was limited to 6,364 articles, future deployments should prioritize continuously expanding the training data with newly labeled articles across more Kenyan outlets, to improve the model's coverage of emerging misinformation narratives.
- iii. Because the current model was trained primarily on English-language content, source and platform owners adopting this approach should invest in extending labeled data collection to Swahili and Sheng-language content, given the multilingual nature of Kenya's digital media landscape noted in Chapters One and Two.
- iv. The monthly retraining and drift-monitoring pipeline built in this study should be maintained and, where resources allow, run more frequently around high-risk periods such as national elections, when the literature and this study's own data collection indicate that fake news volumes rise sharply.
- v. Regulators and stakeholders such as the Independent Electoral and Boundaries Commission (IEBC) and media councils should be engaged with the model's outputs and its documented precision and recall figures, so that any policy or moderation decisions informed by automated detection are grounded in a clear understanding of the model's demonstrated strengths and error rates.
- vi. Because the model currently focuses on public news articles and does not extend to private messaging platforms such as WhatsApp, organizations concerned with misinformation on closed platforms should explore complementary detection or reporting mechanisms suited to that context.
- vii. Given the model's reliance on stylistic and lexical features, it should be paired with human review for borderline or low-confidence predictions, rather than used as a fully automated arbiter of what is fake, to guard against the false positives and false negatives inherent in any classifier of this kind.

5.5 Suggestions for Future Work

While this study successfully developed, trained and deployed a fake news detection model for the Kenyan context, several avenues remain open for future research. First, future work could expand data collection to include private and semi-private messaging platforms such as WhatsApp and Telegram groups,

which this study's scope did not cover but which the literature identifies as significant channels for the spread of fake news in Kenya.

Second, given that the current model relies on TF-IDF combined with hand-crafted stylistic features and a linear classifier (Logistic Regression, with Random Forest and XGBoost evaluated as comparators), future studies should benchmark this approach against additional classical baselines and against transformer-based language models fine-tuned on Kenyan news data, under a shared evaluation protocol, to establish whether the added complexity of such models yields a meaningful improvement over the 95.31% accuracy achieved here.

Third, a Kenyan-context-aware linguistic feature representation was not developed within the scope of this study; the eight hand-crafted features used here are largely generic stylistic indicators rather than markers specific to Kenyan digital news. Future work should investigate linguistic characteristics distinctive to the Kenyan media environment, such as code-switching between English, Swahili and Sheng, locally specific rhetorical patterns, and platform-specific conventions, and incorporate them as an additional feature group alongside TF-IDF and the existing stylistic features. Fourth, this study did not run a formal ablation study isolating the contribution of TF-IDF, the generic stylistic features, and any future Kenyan-context-aware linguistic features. Once such features are developed, future work should evaluate each feature group separately and in combination, to quantify how much of the model's performance each component contributes and to test whether a Kenyan-context-aware representation offers a measurable advantage over the conventional feature set used here. Fifth, while this study took care to prevent data leakage through a stratified 80/20 train-test split and separated synthetic training rows from the fully verified validation and test sets, a more formal audit of dataset composition, labeling consistency, deduplication and leakage across sources would strengthen confidence in the reported results and is recommended before the approach is extended to new data.

Sixth, future work should formally incorporate a feedback loop, such as active learning or human-in-the-loop verification, so that corrections from fact-checkers or users can be systematically fed back into the monthly retraining cycle already established in this study, rather than relying solely on newly scraped and labeled articles.

Seventh, since this study's dataset and features were developed specifically for English-language Kenyan news, future research should examine how the same approach performs when extended to Swahili and Sheng content, and whether the model or its feature set needs to be adapted for other East African media contexts.

Finally, further studies could examine the model's performance over a longer deployment period spanning a general election cycle, to empirically measure how quickly its accuracy degrades between monthly retraining cycles and to refine the drift-detection thresholds established in this study.

ACKNOWLEDGEMENT

I would like to extend my heartfelt gratitude to everyone who supported and contributed to the completion of this study on detecting fake news using machine learning in Kenya. My deepest appreciation goes to my supervisors, Dr. Wakoli and Dr. Ngari, whose guidance, insights, and encouragement were invaluable throughout the research process. I also want to acknowledge the support of my research team and colleagues, whose feedback and expertise significantly enhanced the quality of this study.

I am particularly grateful to everyone involved in the data collection to the data analysis processes. Their cooperation was essential, and I appreciate everyone who was involved in this whole process. Additionally, I would like to thank my family and friends for their unwavering love and understanding, which motivated me to persevere with this work. Lastly, I am thankful to all those who provided the materials and conditions necessary for conducting this research. Your efforts have been crucial in achieving the objectives of this study and advancing the understanding of machine learning in Kenya.

REFERENCES

- [1]. Adjinn-Tettey, T. D. (2022). Combating fake news, disinformation, and misinformation: Experimental evidence for media literacy education. *Cogent Arts & Humanities*, 9(1), Article 2037229. <https://doi.org/10.1080/23311983.2022.2037229>
- [2]. Ahmed, A. A. A., Aljabouh, A., Donepudi, P. K., & Choi, M. S. (2021). *Detecting fake news using machine learning: A systematic literature review*. arXiv. <https://arxiv.org/abs/2102.04458>
- [3]. Ahmed, H., Traore, I., & Saad, S. (2017). Detection of online fake news using n-gram analysis and machine learning techniques. In I. Traore, I. Woungang, & A. Awad (Eds.), *Intelligent, secure, and dependable systems in distributed and cloud environments* (pp. 127–138). Springer. https://doi.org/10.1007/978-3-319-69155-8_9
- [4]. Albahar, M. (2021). A hybrid model for fake news detection: Leveraging news content and user comments in fake news. *IET Information Security*, 15(2), 169–177. <https://doi.org/10.1049/ise2.12021>

- [5]. Alghamdi, J., Lin, Y., & Luo, S. (2022). A comparative study of machine learning and deep learning techniques for fake news detection. *Information*, 13(12), 576. <https://doi.org/10.3390/info13120576>
- [6]. Allcott, H., & Gentzkow, M. (2017). Social media and fake news in the 2016 election. *Journal of Economic Perspectives*, 31(2), 211–236. <https://doi.org/10.1257/jep.31.2.211>
- [7]. Aslam, N., Khan, I. U., Alotaibi, F. S., Aldaej, L. A., & Aldubaikil, A. K. (2021). Fake detect: A deep learning ensemble model for fake news detection. *Complexity*, 2021, Article 5557784. <https://doi.org/10.1155/2021/5557784>
- [8]. Braşoveanu, A. M. P., & Andonie, R. (2019). Semantic fake news detection: A machine learning perspective. In I. Rojas, G. Joya, & A. Català (Eds.), *Advances in Computational Intelligence: 15th International Work-Conference on Artificial Neural Networks, IWANN 2019, Proceedings, Part I* (pp. 656–667). Springer. https://doi.org/10.1007/978-3-030-20521-8_54
- [9]. Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5–32. <https://doi.org/10.1023/A:1010933404324>
- [10]. Busioc, C., Ruseti, S., & Dascalu, M. (2020). A literature review of NLP approaches to fake news detection and their applicability to Romanian-language news analysis. *Revista Transilvania*, (10).
- [11]. Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785–794). ACM. <https://doi.org/10.1145/2939672.2939785>
- [12]. Cheruiyot, D., & Ferrer-Conill, R. (2018). "Fact-checking Africa": Epistemologies, data and the expansion of journalistic discourse. *Digital Journalism*, 6(8), 964–975. <https://doi.org/10.1080/21670811.2018.1493940>
- [13]. Choraś, M., Demestichas, K., Gielczyk, A., Herrero, Á., Ksieniewicz, P., Remoundou, K., Urda, D., & Woźniak, M. (2021). Advanced machine learning techniques for fake news (online disinformation) detection: A systematic mapping study. *Applied Soft Computing*, 101, Article 107050. <https://doi.org/10.1016/j.asoc.2020.107050>
- [14]. Choudhary, M., Jha, S., Saxena, D., & Singh, A. K. (2021). A review of fake news detection methods using machine learning. In *2021 2nd International Conference for Emerging Technology (INCET)* (pp. 1–5). IEEE. <https://doi.org/10.1109/INCET51464.2021.9456358>
- [15]. Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3), 273–297. <https://doi.org/10.1007/BF00994018>
- [16]. Das, M., Kamalanathan, S., & Alphonse, P. (2021). A comparative study on TF-IDF feature weighting method and its analysis using unstructured dataset. In *Proceedings of COLINS-2021: 5th International Conference on Computational Linguistics and Intelligent Systems*.
- [17]. Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5), 1189–1232. <https://doi.org/10.1214/aos/1013203451>
- [18]. Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M., & Bouchachia, A. (2014). A survey on concept drift adaptation. *ACM Computing Surveys*, 46(4), Article 44. <https://doi.org/10.1145/2523813>
- [19]. Government of Kenya. (2019). *The Data Protection Act, 2019* (No. 24 of 2019). Kenya Gazette Supplement No. 181.
- [20]. Granik, M., & Mesyura, V. (2017). Fake news detection using naive Bayes classifier. In *2017 IEEE First Ukraine Conference on Electrical and Computer Engineering (UKRCON)* (pp. 900–903). IEEE. <https://doi.org/10.1109/UKRCON.2017.8100379>
- [21]. Hakak, S., Alazab, M., Khan, S., Gadekallu, T. R., Maddikunta, P. K. R., & Khan, W. Z. (2021). An ensemble machine learning approach through effective feature extraction to classify fake news. *Future Generation Computer Systems*, 117, 47–58. <https://doi.org/10.1016/j.future.2020.11.022>
- [22]. Hansraj, A., Adeliyi, T. T., & Wing, J. (2021). Detection of online fake news using blending ensemble learning. *Scientific Programming*, 2021.
- [23]. Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS Quarterly*, 28(1), 75–105. <https://doi.org/10.2307/25148625>
- [24]. Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>
- [25]. Hutto, C., & Gilbert, E. (2014). VADER: A parsimonious rule-based model for sentiment analysis of social media text. *Proceedings of the International AAAI Conference on Web and Social Media*, 8(1), 216–225. <https://doi.org/10.1609/icwsm.v8i1.14550>
- [26]. International Fact-Checking Network. (2022). *Code of principles*. Poynter Institute. <https://ifcncodeofprinciples.poynter.org/>
- [27]. Islam, M. R., Liu, S., Wang, X., & Xu, G. (2020). Deep learning for misinformation detection on online social networks: A survey and new perspectives. *Social Network Analysis and Mining*, 10(1), Article 82. <https://doi.org/10.1007/s13278-020-00696-x>

- [28]. Islam, N., Shaikh, A., Qaiser, A., Asiri, Y., Almakdi, S., Sulaiman, A., Rajab, K., & Babar, S. A. (2021). Ternion: An autonomous model for fake news detection. *Applied Sciences*, 11(19), 9292. <https://doi.org/10.3390/app11199292>
- [29]. Khan, J. Y., Khondaker, M. T. I., Afroz, S., Uddin, G., & Iqbal, A. (2021). A benchmark study of machine learning models for online fake news detection. *Machine Learning with Applications*, 4, Article 100032. <https://doi.org/10.1016/j.mlwa.2021.100032>
- [30]. Khanam, Z., Alwasel, B. N., Sirafi, H., & Rashid, M. (2021). Fake news detection using machine learning approaches. *IOP Conference Series: Materials Science and Engineering*, 1099, Article 012040. <https://doi.org/10.1088/1757-899X/1099/1/012040>
- [31]. Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence* (Vol. 2, pp. 1137–1143). Morgan Kaufmann.
- [32]. Korkmaz, T., Çetinkaya, A., Aydın, H., & Barışkan, M. A. (2021). Analysis of whether news on the internet is real or fake by using deep learning methods and the TF-IDF algorithm. *International Advanced Researches and Engineering Journal*, 5(1), 31–41. <https://doi.org/10.35860/iaej.779019>
- [33]. Madrid-Morales, D., Wasserman, H., Gondwe, G., Ndlovu, K., Sikanku, E., Tully, M., & Uzuegbunam, C. (2021). Motivations for sharing misinformation: A comparative study in six sub-Saharan African countries. *International Journal of Communication*, 15, 1200–1219.
- [34]. Madung, O. (2022). *Opaque and overstretched, part 2: How platforms failed to curb misinformation during the Kenyan 2022 election*. Mozilla Foundation. <https://www.mozillafoundation.org/en/blog/fellow-research-platforms-failed-to-curb-misinformation-after-kenyan-election/>
- [35]. Maweu, J. M. (2019). "Fake elections"? Cyber propaganda, disinformation and the 2017 general elections in Kenya. *African Journalism Studies*, 40(4), 62–76. <https://doi.org/10.1080/23743670.2020.1719858>
- [36]. Meel, P., & Vishwakarma, D. K. (2020). Fake news, rumor, information pollution in social media and web: A contemporary survey of state-of-the-arts, challenges and opportunities. *Expert Systems with Applications*, 153, Article 112986. <https://doi.org/10.1016/j.eswa.2019.112986>
- [37]. Mishra, S., Shukla, P., & Agarwal, R. (2022). Analyzing machine learning enabled fake news detection techniques for diversified datasets. *Wireless Communications and Mobile Computing*, 2022.
- [38]. Molina, M. D., Sundar, S. S., Le, T., & Lee, D. (2021). "Fake news" is not simply false information: A concept explication and taxonomy of online content. *American Behavioral Scientist*, 65(2), 180–212. <https://doi.org/10.1177/0002764219878224>
- [39]. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.
- [40]. Pennycook, G., & Rand, D. G. (2021). The psychology of fake news. *Trends in Cognitive Sciences*, 25(5), 388–402. <https://doi.org/10.1016/j.tics.2021.02.007>
- [41]. Portland Communications, & GeoPoll. (2017). *The reality of fake news in Kenya*. Portland Communications.
- [42]. Rao, V. C. S., et al. (2021). Logistic regression versus XGBoost: Machine learning for counterfeit news detection. In *2021 Second International Conference on Smart Technologies in Computing, Electrical and Electronics (ICSTCEE)*. IEEE. <https://doi.org/10.1109/ICSTCEE54422.2021.9708587>
- [43]. Reis, J. C. S., Correia, A., Murai, F., Veloso, A., Benevenuto, F., & Cambria, E. (2019). Supervised learning for fake news detection. *IEEE Intelligent Systems*, 34(2), 76–81. <https://doi.org/10.1109/MIS.2019.2899143>
- [44]. Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5), 513–523. [https://doi.org/10.1016/0306-4573\(88\)90021-0](https://doi.org/10.1016/0306-4573(88)90021-0)
- [45]. Shu, K., Sliva, A., Wang, S., Tang, J., & Liu, H. (2017). Fake news detection on social media: A data mining perspective. *ACM SIGKDD Explorations Newsletter*, 19(1), 22–36. <https://doi.org/10.1145/3137597.3137600>
- [46]. Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, 45(4), 427–437. <https://doi.org/10.1016/j.ipm.2009.03.002>
- [47]. Tully, M., Madrid-Morales, D., Wasserman, H., Gondwe, G., & Ireri, K. (2022). Who is responsible for stopping the spread of misinformation? Examining audience perceptions of responsibilities and responses in six sub-Saharan African countries. *Digital Journalism*, 10(5), 679–697. <https://doi.org/10.1080/21670811.2021.1916358>

- [48]. Varma, R., Verma, Y., Vijayvargiya, P., & Churi, P. P. (2021). A systematic survey on deep learning and machine learning approaches of fake news detection in the pre- and post-COVID-19 pandemic. *International Journal of Intelligent Computing and Cybernetics*, 14(4), 617–638. <https://doi.org/10.1108/IJICC-04-2021-0069>
- [49]. Vosoughi, S., Roy, D., & Aral, S. (2018). The spread of true and false news online. *Science*, 359(6380), 1146–1151. <https://doi.org/10.1126/science.aap9559>
- [50]. Wang, W. Y. (2017). "Liar, liar pants on fire": A new benchmark dataset for fake news detection. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics* (Vol. 2, pp. 422–426). Association for Computational Linguistics. <https://doi.org/10.18653/v1/P17-2067>
- [51]. Zhou, X., & Zafarani, R. (2020). A survey of fake news: Fundamental theories, detection methods, and opportunities. *ACM Computing Surveys*, 53(5), Article 109. <https://doi.org/10.1145/3395046>
- [52]. Zhou, X., Jain, A., Phoha, V. V., & Zafarani, R. (2020). Fake news early detection: A theory-driven model. *Digital Threats: Research and Practice*, 1(2), 1–25. <https://doi.org/10.1145/3377478>

APPENDICES

Appendix 1: The code used to train, validate and test the model

```
# Kenyan Fake News Detection v2 — Verified + Synthetic Data
#
# What changed since the last version
#
# This dataset adds **1,250 synthetic FAKE headlines**, written in ordinary
# news-headline style (not fact-checker prose), specifically to weaken the
# "source predicts label" shortcut found in the verified-only data (where
# every FAKE article came from Africa Check/PesaCheck and every REAL article
# came from mainstream outlets).
#
# Key design choices, carried through this script:
# - **Synthetic rows are used in TRAINING ONLY** — validation and test are
#   100% verified, real articles.
# - **Synthetic rows are down-weighted** (`training_weight = 0.25`) during
#   fitting.
# - Text field is `claim_text` (headline-level) — no full article bodies
#   in this dataset.

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import re
from collections import Counter
from wordcloud import WordCloud

from sklearn.model_selection import train_test_split, cross_val_score, GridSearchCV
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from xgboost import XGBClassifier
from sklearn.metrics import (
    accuracy_score, precision_score, recall_score, f1_score,
    classification_report, confusion_matrix, ConfusionMatrixDisplay
)
from nltk.sentiment.vader import SentimentIntensityAnalyzer
import textstat
from scipy.sparse import hstack, csr_matrix

import nltk
nltk.download('vader_lexicon', quiet=True)
sia = SentimentIntensityAnalyzer()
```

```

pd.set_option('display.max_colwidth', 150)

# Data load and cleaning
#
df = pd.read_csv('kenya_fake_news_combined_with_synthetic.csv') # <- update path
print(df.head())

# labeling: 0 -> Real, 1 -> Fake
# `is_synthetic` marks the 1,250 synthetic FAKE examples. `training_weight`
# carries the 0.25 down-weight for synthetic rows (1.0 for verified rows).
df['label'] = (df['class_name'] == 'FAKE').astype(int)
df['claim_text'] = df['claim_text'].fillna("")
df = df[df['claim_text'].str.len() > 0].reset_index(drop=True)

# Class + synthetic composition
print(f"Total rows: {len(df)}")
print(df.groupby(['is_synthetic', 'class_name']).size())

fig, axes = plt.subplots(1, 2, figsize=(12, 5))
df['class_name'].value_counts().plot(kind='pie', autopct='%1.1f%%', ax=axes[0], colors=['#2ecc71', '#e74c3c'])
axes[0].set_ylabel("")
axes[0].set_title('Overall class balance')

df['is_synthetic'].value_counts().rename({False:'Verified', True:'Synthetic'}).plot(
    kind='pie', autopct='%1.1f%%', ax=axes[1], colors=['#3498db', '#f39c12'])
axes[1].set_ylabel("")
axes[1].set_title('Verified vs Synthetic')
plt.tight_layout()
plt.show()

# Nearly half of the synthetic FAKE examples share a literal opening phrase.
prefix_counts = df[df['is_synthetic']][['claim_text']].str.startswith('Government confirms').sum()
total_synth = df['is_synthetic'].sum()
print(f'Synthetic rows starting with literal "Government confirms": {prefix_counts} / {total_synth} '
      f'({prefix_counts/total_synth*100:.1f}%)')

#
# Preprocessing
#
def preprocess_text(text):
    text = str(text).lower()
    text = re.sub(r'http\S+|www\S+', "", text)
    text = re.sub(r'd+', "", text)
    text = re.sub(r'^a-z\s', "", text)
    text = re.sub(r's+', ' ', text).strip()
    return text

df["clean_text"] = df["claim_text"].apply(preprocess_text)
print(df[["claim_text", "clean_text"]].sample(5))

# Lean engineered features (headline-level: style + sentiment)
def style_features(text):
    words = text.split()
    n_words = max(len(words), 1)
    n_chars = len(text)
    avg_word_len = np.mean([len(w) for w in words]) if words else 0
    n_exclaim = text.count('!')
    n_question = text.count('?')

```

```

n_upper = sum(1 for c in text if c.isupper())
cap_ratio = n_upper / max(n_chars, 1)
try:
    readability = textstat.flesch_reading_ease(text)
except Exception:
    readability = 0.0
scores = sia.polarity_scores(text)
return [n_words, n_chars, avg_word_len, n_exclaim, n_question, cap_ratio,
        readability, scores['neg'], scores['neu'], scores['pos'], scores['compound']]

engineered = np.array([style_features(t) for t in df["claim_text"]])
engineered_cols = ["n_words", "n_chars", "avg_word_len", "n_exclaim", "n_question",
                  "cap_ratio", "readability", "sent_neg", "sent_neu", "sent_pos", "sent_compound"]
print(engineered.shape)

#
# Train / Validation / Test Split
#
# Verified rows only are split into train/val/test (70/15/15, stratified).
# All synthetic rows go into training only, carrying their 0.25 weight.
verified = df[df["is_synthetic"] == False].copy()
synthetic = df[df["is_synthetic"] == True].copy()

idx_train_v, idx_temp_v = train_test_split(
    verified.index, test_size=0.30, random_state=42, stratify=verified["label"]
)
idx_val_v, idx_test_v = train_test_split(
    idx_temp_v, test_size=0.50, random_state=42, stratify=verified.loc[idx_temp_v, "label"]
)

train_idx = list(idx_train_v) + list(synthetic.index)
val_idx = list(idx_val_v)
test_idx = list(idx_test_v)

print(f"Train: {len(train_idx)} rows (verified={len(idx_train_v)}, synthetic={len(synthetic)})")
print(f"Val (verified only): {len(val_idx)} rows")
print(f"Test (verified only): {len(test_idx)} rows")

sample_weight_train = df.loc[train_idx, "training_weight"].fillna(1.0).values

#
# Word Cloud & Pattern Analysis
#
fake_text = " ".join(df.loc[train_idx][df.loc[train_idx, 'label']==1]['clean_text'])
real_text = " ".join(df.loc[train_idx][df.loc[train_idx, 'label']==0]['clean_text'])

fig, axes = plt.subplots(1, 2, figsize=(18, 7))
wc_fake = WordCloud(width=900, height=450, background_color='white', colormap='Reds',
max_words=100).generate(fake_text)
wc_real = WordCloud(width=900, height=450, background_color='white', colormap='Greens',
max_words=100).generate(real_text)
axes[0].imshow(wc_fake, interpolation='bilinear'); axes[0].set_title('Fake (Verified + Synthetic) — Most
Frequent Words'); axes[0].axis('off')
axes[1].imshow(wc_real, interpolation='bilinear'); axes[1].set_title('Real — Most Frequent Words');
axes[1].axis('off')
plt.tight_layout(); plt.show()

def top_words(texts, n=15):
    return Counter(" ".join(texts).split()).most_common(n)

```

```

fake_top = top_words(df.loc[train_idx][df.loc[train_idx,'label']==1]['clean_text'])
real_top = top_words(df.loc[train_idx][df.loc[train_idx,'label']==0]['clean_text'])

fig, axes = plt.subplots(1, 2, figsize=(16, 6))
w, c = zip(*fake_top)
axes[0].barh(w[:-1], c[:-1], color='#e74c3c', edgecolor='black'); axes[0].set_title("Top 15 Words — Fake");
axes[0].set_xlabel('Frequency')
w, c = zip(*real_top)
axes[1].barh(w[:-1], c[:-1], color='#2ecc71', edgecolor='black'); axes[1].set_title("Top 15 Words — Real");
axes[1].set_xlabel('Frequency')
plt.tight_layout(); plt.show()

# Note: fact-check vocabulary ("fabricated", "doctored", "does", "isn't")
# on the Fake side is carried over from the verified Africa Check/PesaCheck
# headlines — discussed further after model training.

#
# TF-IDF Feature Extraction
#
tfidf_vectorizer = TfidfVectorizer(max_features=6000, ngram_range=(1, 2), min_df=2,
                                   max_df=0.8, stop_words='english', sublinear_tf=True)

X_train_tfidf = tfidf_vectorizer.fit_transform(df.loc[train_idx, 'clean_text'])
X_val_tfidf = tfidf_vectorizer.transform(df.loc[val_idx, 'clean_text'])
X_test_tfidf = tfidf_vectorizer.transform(df.loc[test_idx, 'clean_text'])

scaler = StandardScaler()
X_train_eng = scaler.fit_transform(engineered[train_idx])
X_val_eng = scaler.transform(engineered[val_idx])
X_test_eng = scaler.transform(engineered[test_idx])

X_train = hstack([X_train_tfidf, csr_matrix(X_train_eng)])
X_val = hstack([X_val_tfidf, csr_matrix(X_val_eng)])
X_test = hstack([X_test_tfidf, csr_matrix(X_test_eng)])

y_train = df.loc[train_idx, 'label'].values
y_val = df.loc[val_idx, 'label'].values
y_test = df.loc[test_idx, 'label'].values

feature_names = np.array(list(tfidf_vectorizer.get_feature_names_out()) + engineered_cols)
print(X_train.shape)

#
# MODEL 1 — Logistic Regression + GridSearch
#
model = LogisticRegression(max_iter=1000, class_weight='balanced')
scores = cross_val_score(model, X_train, y_train, cv=3, scoring='f1_weighted',
                          params={'sample_weight': sample_weight_train})
print("LR CV Scores:", scores, "Mean:", scores.mean())

param_grid_lr = {'C': [0.1, 1, 10], 'penalty': ['l2'], 'solver': ['lbfgs']}
grid_lr = GridSearchCV(LogisticRegression(max_iter=1000, class_weight='balanced'),
                       param_grid_lr, cv=5, scoring='f1_weighted', n_jobs=-1)
grid_lr.fit(X_train, y_train, sample_weight=sample_weight_train)
print(f"LR Best Params: {grid_lr.best_params_} Best CV: {grid_lr.best_score_:.4f}")

best_lr = grid_lr.best_estimator_
y_pred_lr = best_lr.predict(X_test)

```

```

lr_acc = accuracy_score(y_test, y_pred_lr)
lr_pre = precision_score(y_test, y_pred_lr)
lr_rec = recall_score(y_test, y_pred_lr)
lr_f1 = f1_score(y_test, y_pred_lr)
lr_f1_macro = f1_score(y_test, y_pred_lr, average='macro')

print(f"LR -> Accuracy: {lr_acc*100:.2f}% Precision: {lr_pre*100:.2f}% Recall: {lr_rec*100:.2f}% "
      f"F1: {lr_f1*100:.2f}% F1-macro: {lr_f1_macro*100:.2f}%")
print(classification_report(y_test, y_pred_lr, target_names=['Real', 'Fake']))

cm_lr = confusion_matrix(y_test, y_pred_lr)
disp = ConfusionMatrixDisplay(cm_lr, display_labels=['Real', 'Fake'])
disp.plot(cmap='Blues')
plt.title('Logistic Regression — Confusion Matrix (verified test set)')
plt.tight_layout(); plt.show()

coef = best_lr.coef_[0]
top_fake_words = feature_names[coef.argsort()[-15:][::-1]]
top_real_words = feature_names[coef.argsort()[0:15]]
fig, axes = plt.subplots(1, 2, figsize=(16, 6))
axes[0].barh(top_fake_words[:-1], coef[coef.argsort()[-15:][::-1]][::-1], color='#e74c3c', edgecolor='black')
axes[0].set_title('Top Features → FAKE'); axes[0].set_xlabel('Coefficient Value')
axes[1].barh(top_real_words[:-1], coef[coef.argsort()[0:15]][::-1], color='#2ecc71', edgecolor='black')
axes[1].set_title('Top Features → REAL'); axes[1].set_xlabel('Coefficient Value')
plt.tight_layout(); plt.show()

#
# MODEL 2 — Random Forest + GridSearch
#
rf_base = RandomForestClassifier(random_state=42, class_weight='balanced', n_jobs=-1)
scores_rf = cross_val_score(rf_base, X_train, y_train, cv=3, scoring='f1_weighted',
                             params={'sample_weight': sample_weight_train})
print("RF CV Scores:", scores_rf, "Mean:", scores_rf.mean())

param_grid_rf = {'n_estimators': [200], 'max_depth': [None, 20]}
grid_rf = GridSearchCV(RandomForestClassifier(random_state=42, class_weight='balanced', n_jobs=-1),
                        param_grid_rf, cv=3, scoring='f1_weighted', n_jobs=-1)
grid_rf.fit(X_train, y_train, sample_weight=sample_weight_train)
print(f"RF Best Params: {grid_rf.best_params_} Best CV: {grid_rf.best_score_:.4f}")

best_rf = grid_rf.best_estimator_
y_pred_rf = best_rf.predict(X_test)

rf_acc = accuracy_score(y_test, y_pred_rf)
rf_pre = precision_score(y_test, y_pred_rf)
rf_rec = recall_score(y_test, y_pred_rf)
rf_f1 = f1_score(y_test, y_pred_rf)
rf_f1_macro = f1_score(y_test, y_pred_rf, average='macro')

print(f"RF -> Accuracy: {rf_acc*100:.2f}% Precision: {rf_pre*100:.2f}% Recall: {rf_rec*100:.2f}% "
      f"F1: {rf_f1*100:.2f}% F1-macro: {rf_f1_macro*100:.2f}%")
print(classification_report(y_test, y_pred_rf, target_names=['Real', 'Fake']))

cm_rf = confusion_matrix(y_test, y_pred_rf)
disp_rf = ConfusionMatrixDisplay(cm_rf, display_labels=['Real', 'Fake'])
disp_rf.plot(cmap='Purples')
plt.title('Random Forest — Confusion Matrix (verified test set)')
plt.tight_layout(); plt.show()

```

```

importances = best_rf.feature_importances_
top_idx = importances.argsort()[-15:][::-1]
plt.figure(figsize=(8, 6))
plt.barh(feature_names[top_idx][::-1], importances[top_idx][::-1], color='#9b59b6', edgecolor='black')
plt.title('Random Forest — Top 15 Feature Importances'); plt.xlabel('Importance')
plt.tight_layout(); plt.show()

#
# MODEL 3 — XGBoost + GridSearch
#
neg, pos = (y_train == 0).sum(), (y_train == 1).sum()
scale_pos_weight = neg / pos
print(f"scale_pos_weight = {scale_pos_weight:.3f}")

xgb_base = XGBClassifier(eval_metric='logloss', random_state=42, n_jobs=-1,
scale_pos_weight=scale_pos_weight)
scores_xgb = cross_val_score(xgb_base, X_train, y_train, cv=3, scoring='f1_weighted',
params={'sample_weight': sample_weight_train})
print("XGB CV Scores:", scores_xgb, "Mean:", scores_xgb.mean())

param_grid_xgb = {'n_estimators': [200], 'max_depth': [4, 6], 'learning_rate': [0.1]}
grid_xgb = GridSearchCV(
XGBClassifier(eval_metric='logloss', random_state=42, n_jobs=-1, scale_pos_weight=scale_pos_weight),
param_grid_xgb, cv=3, scoring='f1_weighted', n_jobs=-1)
grid_xgb.fit(X_train, y_train, sample_weight=sample_weight_train)
print(f"XGB Best Params: {grid_xgb.best_params_} Best CV: {grid_xgb.best_score_:.4f}")

best_xgb = grid_xgb.best_estimator_
y_pred_xgb = best_xgb.predict(X_test)

xgb_acc = accuracy_score(y_test, y_pred_xgb)
xgb_pre = precision_score(y_test, y_pred_xgb)
xgb_rec = recall_score(y_test, y_pred_xgb)
xgb_f1 = f1_score(y_test, y_pred_xgb)
xgb_f1_macro = f1_score(y_test, y_pred_xgb, average='macro')

print(f"XGB -> Accuracy: {xgb_acc*100:.2f}% Precision: {xgb_pre*100:.2f}% Recall: {xgb_rec*100:.2f}%
"
f"F1: {xgb_f1*100:.2f}% F1-macro: {xgb_f1_macro*100:.2f}%")
print(classification_report(y_test, y_pred_xgb, target_names=['Real', 'Fake']))

cm_xgb = confusion_matrix(y_test, y_pred_xgb)
disp_xgb = ConfusionMatrixDisplay(cm_xgb, display_labels=['Real', 'Fake'])
disp_xgb.plot(cmap='Oranges')
plt.title('XGBoost — Confusion Matrix (verified test set)')
plt.tight_layout(); plt.show()

importances_xgb = best_xgb.feature_importances_
top_idx_xgb = importances_xgb.argsort()[-15:][::-1]
plt.figure(figsize=(8, 6))
plt.barh(feature_names[top_idx_xgb][::-1], importances_xgb[top_idx_xgb][::-1], color='#e67e22',
edgecolor='black')
plt.title('XGBoost — Top 15 Feature Importances'); plt.xlabel('Importance')
plt.tight_layout(); plt.show()

#
# Model Comparison
#

```

```

results = pd.DataFrame([
    {'Model': 'Logistic Regression', 'Accuracy': round(lr_acc*100,2), 'Precision': round(lr_pre*100,2),
    'Recall': round(lr_rec*100,2), 'F1-Score': round(lr_f1*100,2), 'F1-Macro': round(lr_f1_macro*100,2)},
    {'Model': 'Random Forest', 'Accuracy': round(rf_acc*100,2), 'Precision': round(rf_pre*100,2),
    'Recall': round(rf_rec*100,2), 'F1-Score': round(rf_f1*100,2), 'F1-Macro': round(rf_f1_macro*100,2)},
    {'Model': 'XGBoost', 'Accuracy': round(xgb_acc*100,2), 'Precision': round(xgb_pre*100,2),
    'Recall': round(xgb_rec*100,2), 'F1-Score': round(xgb_f1*100,2), 'F1-Macro': round(xgb_f1_macro*100,2)},
])
print(results)

metrics = ['Accuracy', 'Precision', 'Recall', 'F1-Score']
x = np.arange(len(metrics))
width = 0.25
fig, ax = plt.subplots(figsize=(12, 6))
ax.bar(x - width, results.iloc[0][metrics].values, width, label='Logistic Regression', color='#3498db',
edgecolor='black')
ax.bar(x, results.iloc[1][metrics].values, width, label='Random Forest', color='#9b59b6', edgecolor='black')
ax.bar(x + width, results.iloc[2][metrics].values, width, label='XGBoost', color='#e67e22', edgecolor='black')
ax.set_xticks(x); ax.set_xticklabels(metrics); ax.set_ylabel('Score (%)')
ax.set_title('Model Comparison — verified-only held-out test set')
ax.legend(); ax.grid(axis='y', linestyle='--', alpha=0.5)
plt.tight_layout(); plt.show()

#
# Leakage & Template-Artifact Check (the most important section — read this)
#
print("Top 20 FAKE-leaning features per model:\n")
coef = best_lr.coef_[0]
print("Logistic Regression:", list(feature_names[coef.argsort()[-20:][::-1]]))
print()
print("Random Forest:", list(feature_names[best_rf.feature_importances_.argsort()[-20:][::-1]]))
print()
print("XGBoost:", list(feature_names[best_xgb.feature_importances_.argsort()[-20:][::-1]]))

# Findings:
# - "government confirms" / "government" / "confirms" rank near the top
# across all three models — traced to the synthetic set, where 47% of
# FAKE examples literally start with "Government confirms." This is a
# template fingerprint, not a real misinformation signal.
# - "fabricated", "doctored", "does not", "false", "isn't", "claiming"
# also rank highly — fact-checker debunking vocabulary from the
# verified Africa Check/PesaCheck headlines, not vocabulary that
# appears in the fake claims themselves.
#
# Conclusion: these results are more trustworthy than a version trained
# without synthetic data (which hit ~99% by fully exploiting the source
# confound), but they are not yet a validated real-world fake-news
# detector. Treat the ~91-95% scores as a diagnostic checkpoint.

#
# Explainability
#
top_fake_words = feature_names[coef.argsort()[-30:][::-1]]
print("Top Fake-News Keywords:")
for word in top_fake_words:
    print(word)

FAKE_PATTERNS = list(top_fake_words)

```

```
best_model_row = results.loc[results['F1-Score'].idxmax()]
model_lookup = {'Logistic Regression': best_lr, 'Random Forest': best_rf, 'XGBoost': best_xgb}
best_model = model_lookup[best_model_row['Model']]
print(f"Best model selected: {best_model_row['Model']} (F1 = {best_model_row['F1-Score']})")
```

```
def explain_prediction(text):
    clean = preprocess_text(text)
    tfidf_vec = tfidf_vectorizer.transform([clean])
    eng_vec = scaler.transform(np.array([style_features(text)]))
    vec = hstack([tfidf_vec, csr_matrix(eng_vec)])

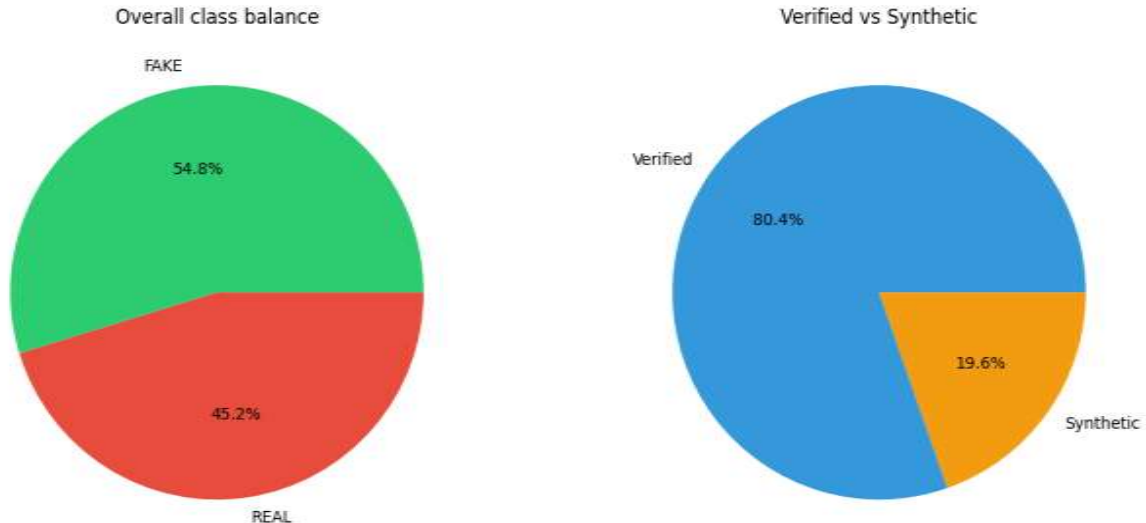
    proba = best_model.predict_proba(vec)[0]
    pred = int(np.argmax(proba))
    prediction = "FAKE" if pred == 1 else "REAL"
    confidence = proba[pred]

    tokens = clean.split()
    keywords = [w for w in tokens if w in FAKE_PATTERNS]

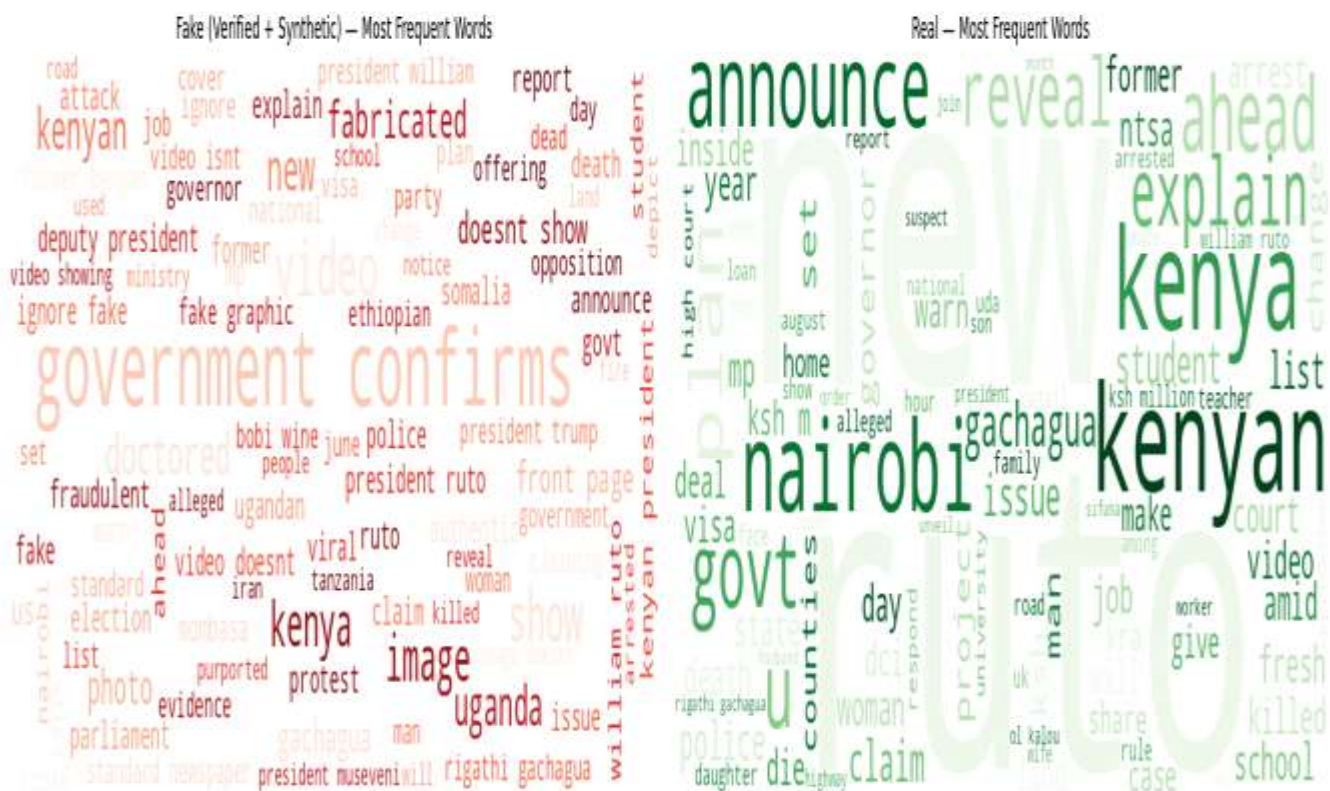
    print(f"Prediction : {prediction}")
    print(f"Confidence : {confidence*100:.2f}%")
    print(f"Matched fake-pattern keywords: {keywords if keywords else 'none'}")
    return prediction, confidence
```

```
explain_prediction("Breaking: You won't believe what the government is hiding!")
explain_prediction("According to Reuters, the Federal Reserve announced a 0.25 percent increase in interest rates to control inflation.")
explain_prediction("Government confirms secret plan exposed by anonymous sources")
```

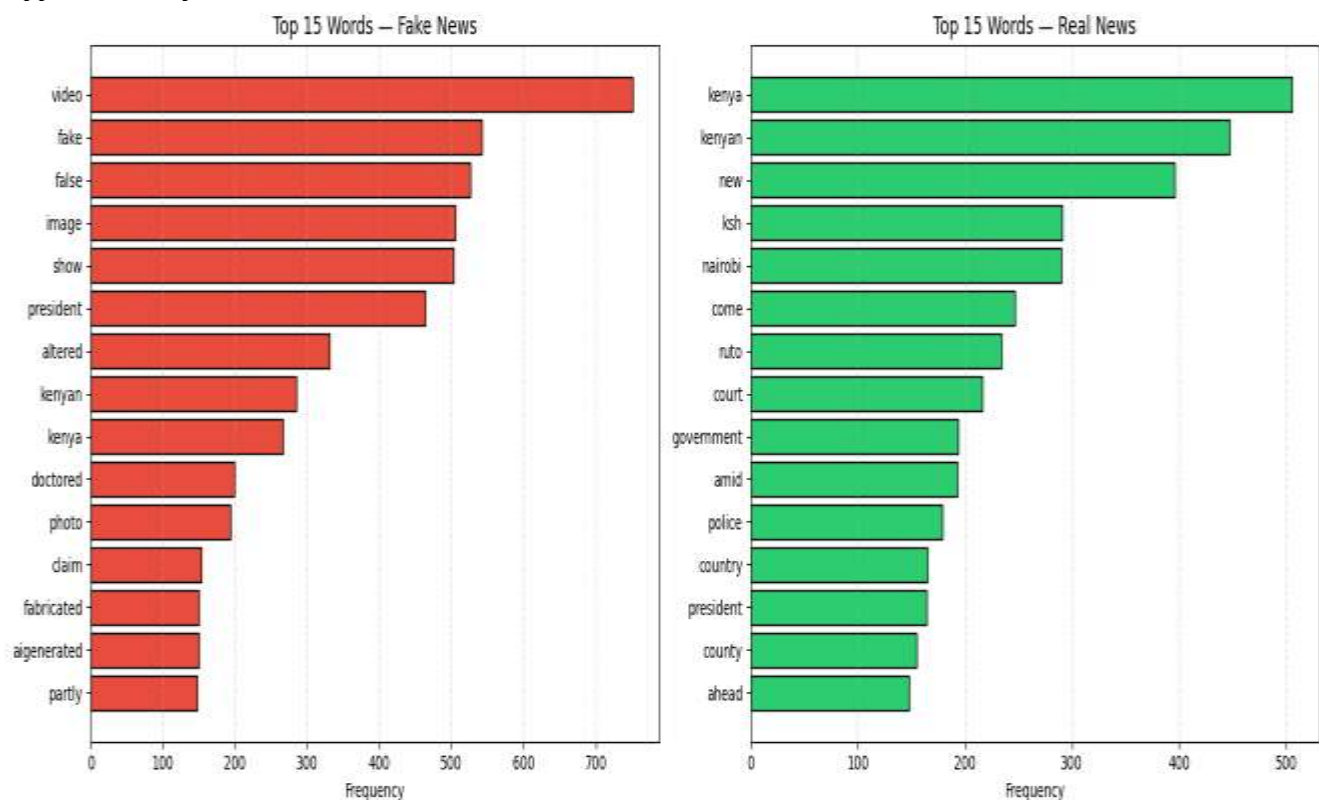
Appendix 2: News Distribution



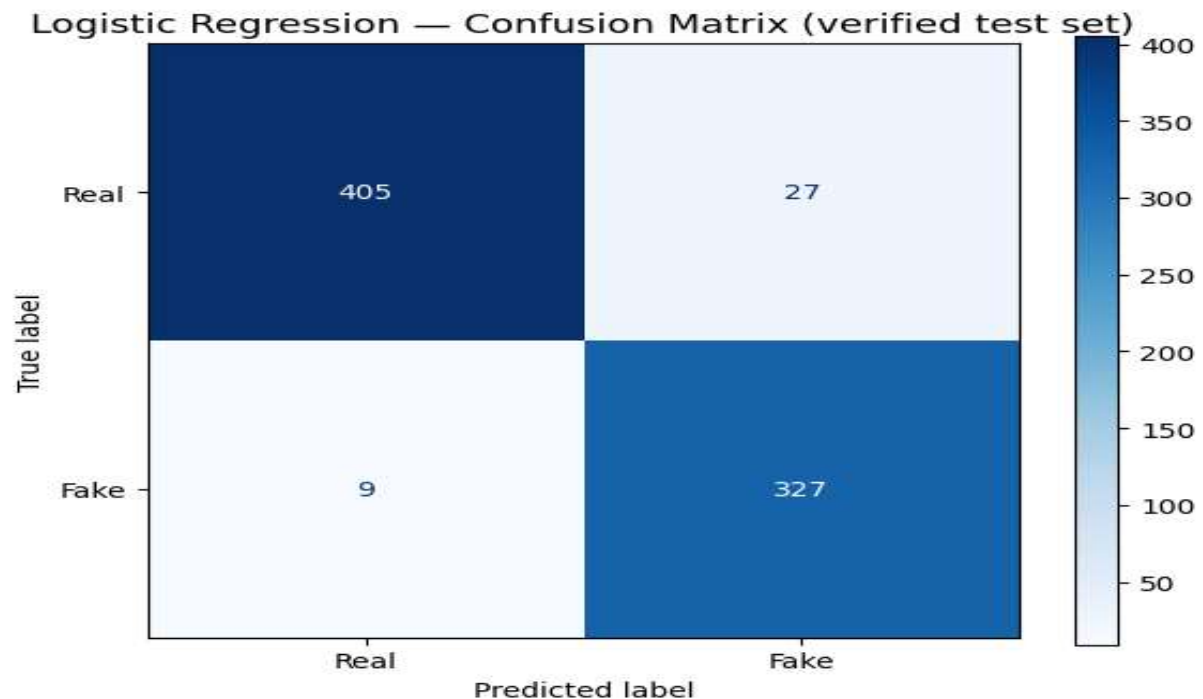
Fake (Verified + Synthetic) — Most Frequent Words



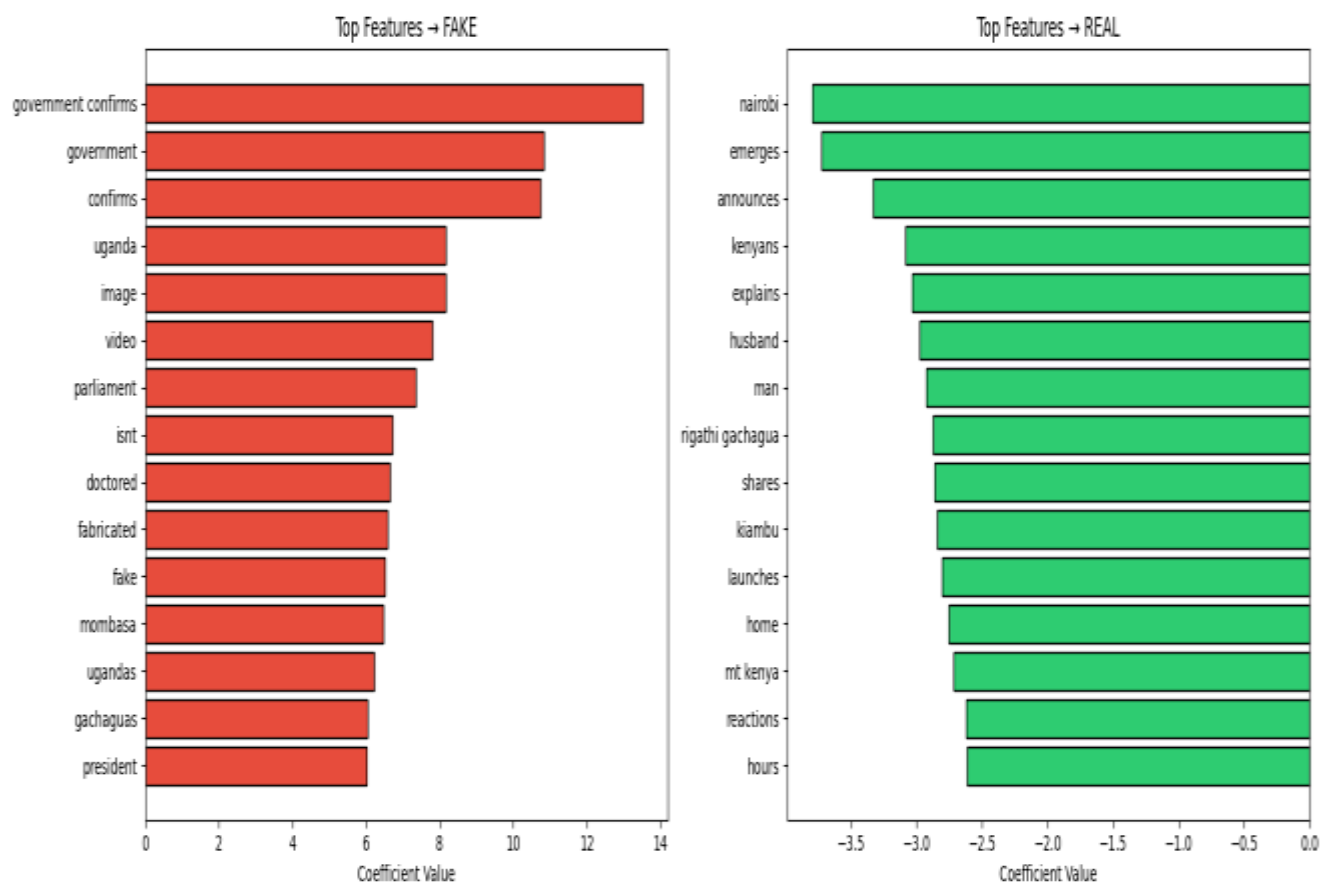
Top 15 Words — Fake M



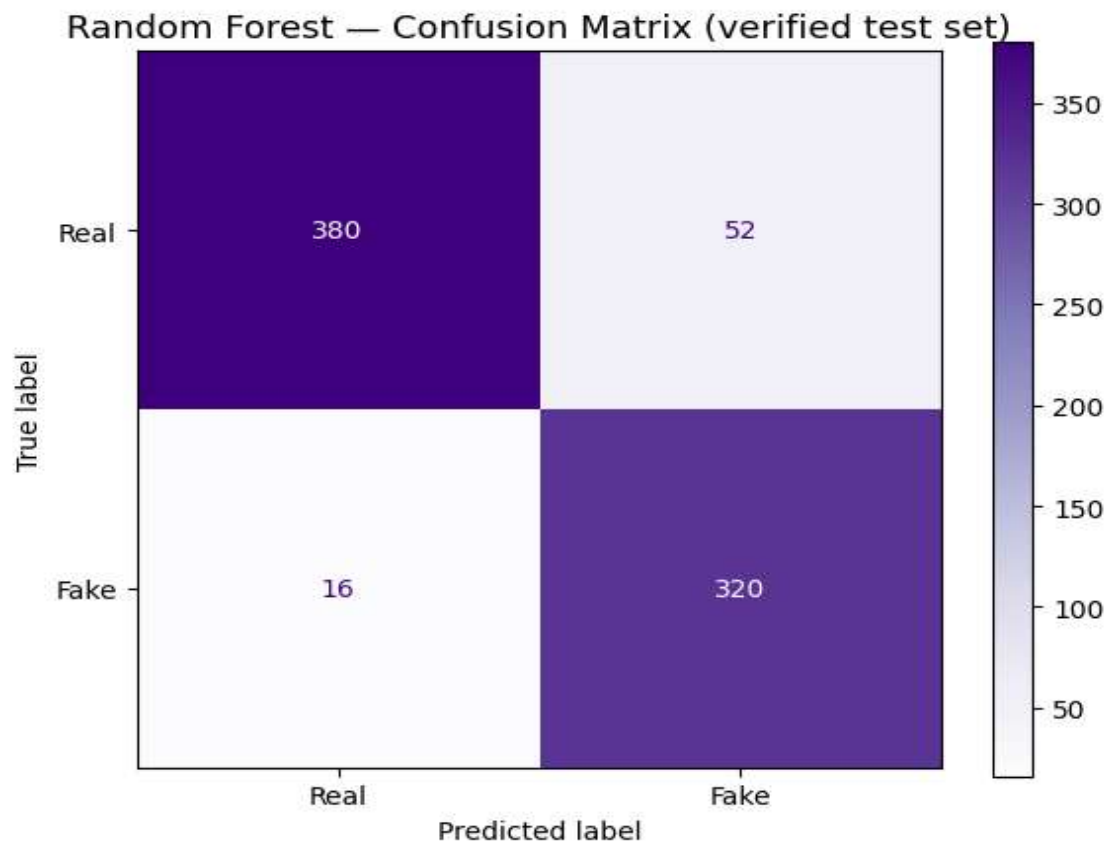
Appendix 5: Logistic Regression- Confusion Matrix



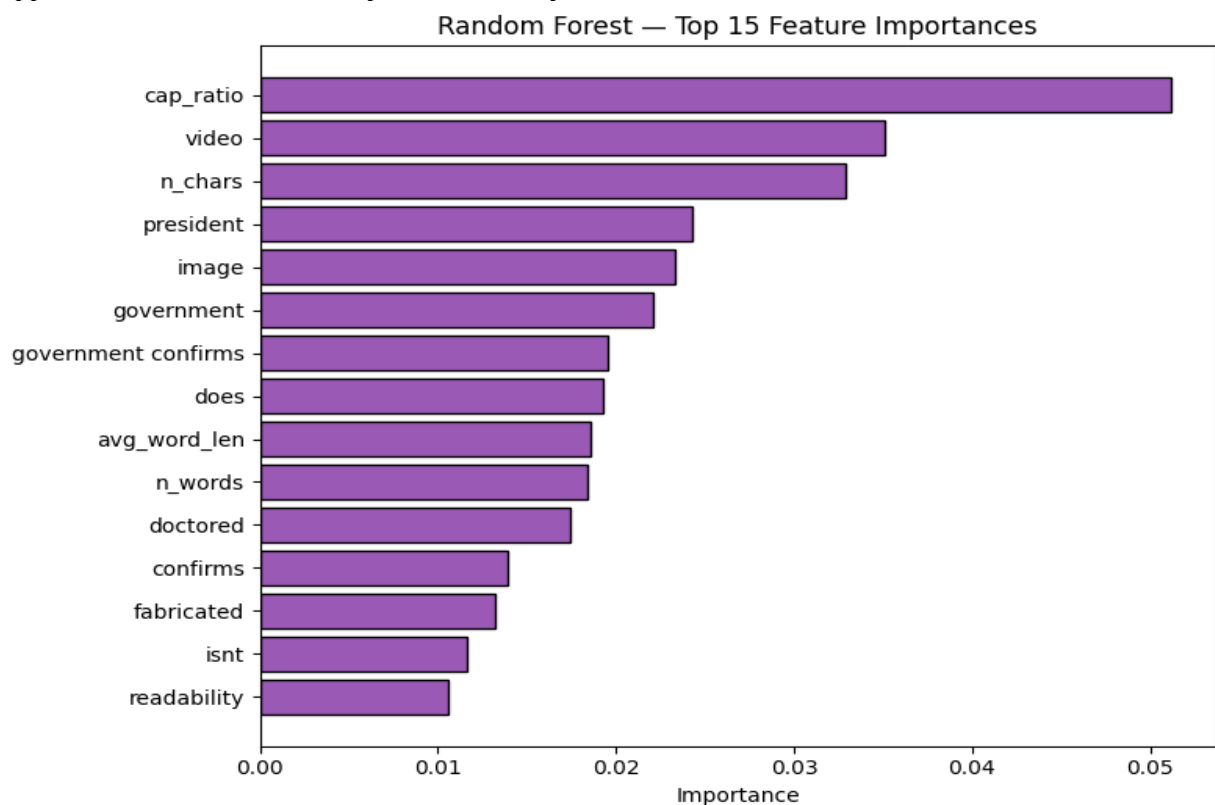
Appendix 6: TOP FEATURES



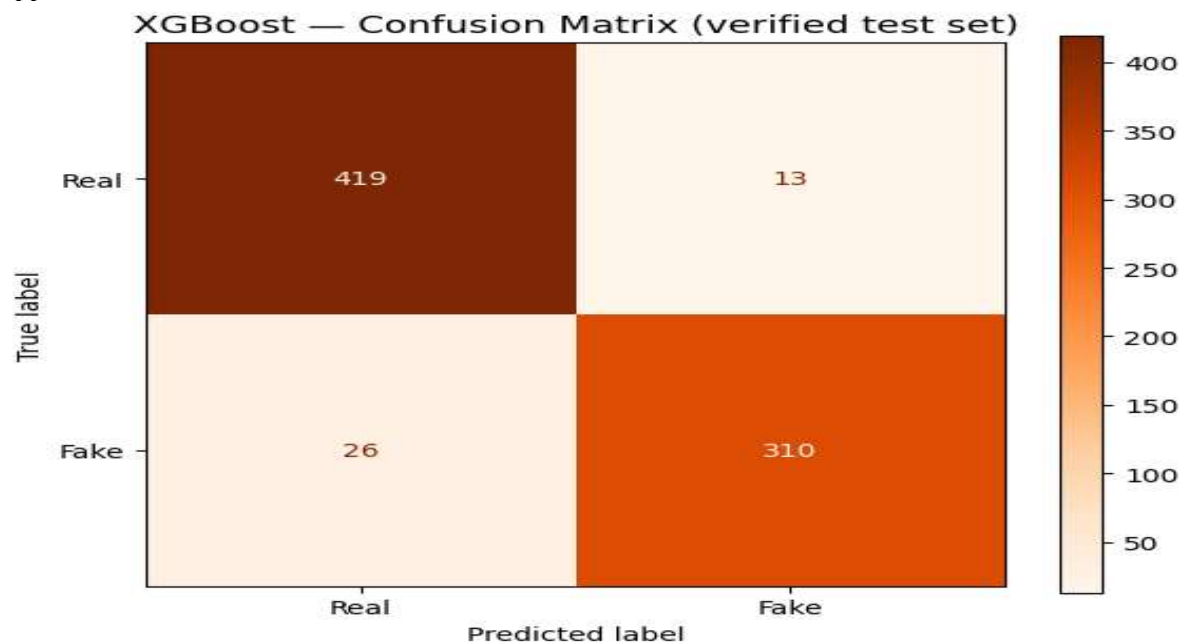
Appendix 7: Random Forest- Confusion Matrix



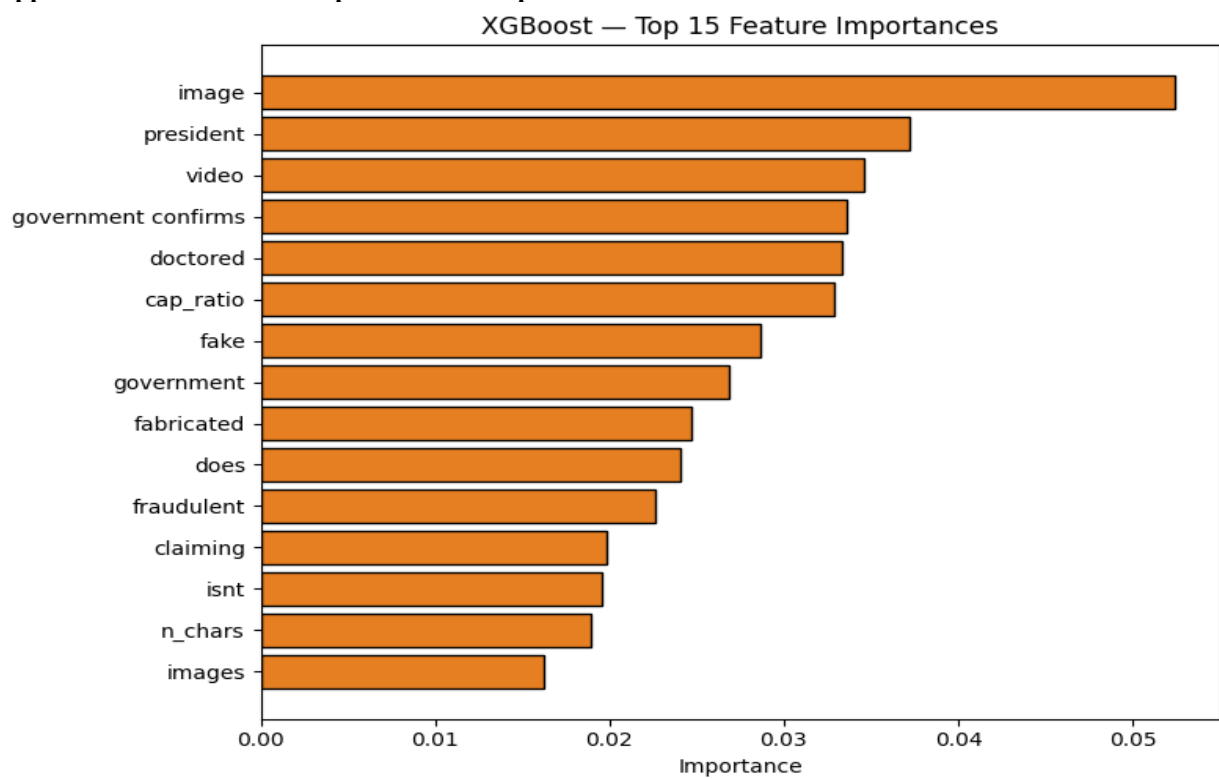
Appendix 8: Random Forest- Top 15 features importances



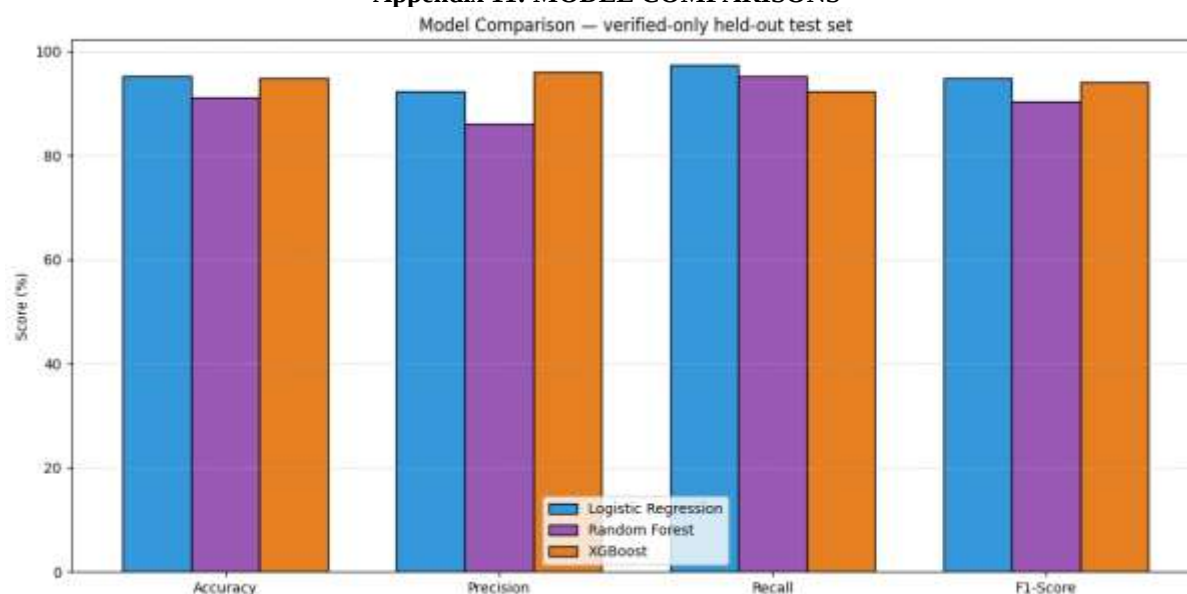
Appendix 9: XGBOOST- Confusion Matrix



Appendix 10: XGBOOST- Top 15 features importances



Appendix 11: MODEL COMPARISONS

**DECLARATION**

I declare that this research proposal is my own original work and has not been submitted for any other degree or qualification at any other institution.

CHEGE KIMONDO JOHN

CIT-323-002/2023

This thesis has been submitted for examination with our approval as University supervisors.

DEDICATION

I dedicate this work to my parents for their unwavering support throughout this whole process. It is during the tough times, that you reminded me to keep going and not to give up even when I had given up. I owe this to you. And to my siblings (Francis, Michael and Elizabeth) I hope this thesis shines a light to you to follow the same path and do the family proud.