J. Electrical Systems 20-11s (2024): 4577-4599

Tanzeela Zarger[1*],
Shobha Lal[2]

Fake news Detection on Social Media Using Machine Learning

Journal of Electrical Systems

Abstract:
The spread of misinformation on the internet and social media has become a growing problem, impacting public opinion and decision-making. In this study, we explore how machine learning can be used to detect and classify fake news more effectively. We gathered a diverse dataset from various online sources and applied a series of preprocessing steps to prepare the data for analysis. This included techniques like tokenization, normalization, removing punctuation and stop words, and lemmatization. To improve the quality of the analysis, we developed an Enhanced Feature Engineering framework for Fake News Detection. This framework combined key features such as TF-IDF, Bag of Words, tweet length, and sentiment analysis, creating a robust dataset for training machine learning models. Among the various models we tested, the ensemble voting classifier stood out for its accuracy and reliability in distinguishing real news from fake. Its strong performance demonstrates the potential of combining multiple algorithms to tackle complex problems like misinformation.
Through this research, we aim to contribute to the ongoing efforts to combat fake news, helping to create a more reliable and trustworthy digital space.

1. Introduction

Social media has revolutionized the way we communicate, share information, and interact with the world. Platforms such as Facebook, Twitter, and Instagram are no longer just tools for personal connection—they have become dominant sources of news and information. However, this digital revolution comes with a significant downside: the rapid proliferation of fake news. Defined as deliberately false or misleading information, fake news thrives on social media due to its viral nature and algorithms that prioritize sensational content over accuracy. In an environment where information spreads at lightning speed, distinguishing fact from fiction has become increasingly difficult.

The consequences of fake news are far-reaching, influencing political elections, deepening social divides, and undermining trust in journalism and public institutions. Social media's unique characteristics exacerbate the problem. Unlike traditional news outlets, these platforms lack stringent editorial oversight, allowing anyone to create and share content without verifying its accuracy. Moreover, the interactive and emotionally driven nature of social media encourages users to share posts impulsively, further fueling the spread of misinformation. This has created an ecosystem where fake news not only survives but often outperforms factual information in reach and engagement.

While traditional methods like fact-checking and content moderation play a role in addressing fake news, they are insufficient for the scale and complexity of social media. The sheer volume of user-generated content makes manual efforts impractical, while automated systems often lack transparency and adaptability. Moreover, fake news creators continuously evolve their tactics to evade detection. This dynamic nature of misinformation demands innovative, scalable, and adaptable solutions that can keep pace with these challenges.

Machine Learning (ML) is being used nowadays in almost every field to provide different solutions using structured and un structured data [1] . It offers a promising approach to combating fake news on social media. By analyzing large volumes of data and identifying complex patterns, machine learning algorithms can process textual, visual, and contextual features to assess the credibility of posts. For instance, a machine learning model might analyze the linguistic structure of a tweet, the reliability of its source, and its overall sentiment to determine its authenticity. This capability to integrate and evaluate diverse features makes machine learning particularly effective in addressing the unique challenges of social media platforms.

Our research focuses on leveraging machine learning techniques to detect textual fake news specifically on social media. Unlike general fake news detection, this study addresses challenges unique to social media, such as short text formats, rapid content dissemination, and user-driven interactions. To begin, we curated a diverse dataset from various platforms, encompassing both genuine and fake news posts. This dataset was preprocessed through techniques like tokenization, punctuation removal, stop word elimination, and sentiment analysis, ensuring the data was clean and ready for model training.

Feature engineering played a critical role in our approach. We combined traditional text-based features like term frequency-inverse document frequency (TF-IDF) and Bag of Words with social media-specific features, including post length, hashtag usage, user engagement metrics, and sentiment polarity. This robust feature set formed the foundation for training machine learning models to distinguish between authentic and fake news. Among the various models tested, ensemble methods—such as voting classifiers—demonstrated superior performance, effectively combining the strengths of multiple algorithms for higher accuracy.

[1]Jayoti Vidhyapeeth Women's university
[2]Jayoti Vidhyapeeth Women's university

The findings of our study highlight the significant potential of machine learning in combating fake news on social media. However, we acknowledge that no solution is perfect. Fake news is a dynamic and evolving challenge, requiring continuous research and innovation. While machine learning enhances detection capabilities, it must be complemented by broader efforts, including user education, platform accountability, and ethical considerations in algorithm design.

By focusing specifically on social media—a primary battleground for misinformation—our re- search aims to contribute meaningful solutions to this pressing issue. The ultimate goal is to create tools that empower users, promote informed decision-making, and restore trust in online information. Through the fusion of technology and social responsibility, we believe it is possible to curb the spread of fake news and foster a healthier, more trustworthy digital environment.

2. Literature Review

”Machine learning is the fuel that powers intelligent systems, enabling them to learn from data, adapt, and make informed decisions.”(Fei-Fei Li) The detection of false news has attracted significant attention from scholars worldwide. Extensive research in social science has been carried out to understand the impact of false news and how individuals respond to it. The use of social media for news consumption has both advantages and disadvantages. Many users prefer social media for news due to its accessibility, user-friendliness, and rapid information dissemination. However, it also serves as a platform for spreading low-quality news containing deliberate misinformation, which can have detrimental effects on society and individuals. Consequently, the study of detecting false information on social media presents distinct challenges that render traditional news media identification algorithms ineffective or impractical. This chapter provides a comprehensive review of false news, including its various phases, methods for detecting false news, feature selections, and the utilization of Machine Learning (ML) ensembles and Deep Learning (DL) techniques.

[2]proposed a method in their study to help users detect fake news and identify misleading content. Their approach involves analyzing past true and false news occurrences to assess the likelihood of specific news items being intentionally deceptive. This research is significant as it addresses a notable challenge in Natural Language Processing (NLP) by not relying on deceitful news datasets for detection purposes. The study focuses on three categories of false news and compares them to instances of legitimate factual reporting, examining the pros and cons of utilizing these categories as a corpus for text analysis and predictive modeling. With the increasing convergence of traditional and online news, ensuring the proper filtration, validation, and verification of internet content has become imperative in the fields of library and information science.

Support mechanisms for detecting and addressing deception in Internet scams were developed by researchers [3]. The study focused on evaluating both legitimate and fraudulent projects present on crowdfunding platforms. Instead of relying solely on static project data, the researchers monitored the dynamic communications that occurred during the funding phases, leading to the successful identification of fraudulent activities. Linguistic cues and content-based signals were also analyzed to aid in fraud detection. The cue selections and feature engineering were based on a combination of communication, computational, psychological, and linguistic theories. The findings of this study have proven beneficial for stakeholders and researchers involved in fraud detection.[4] put forth a method for detecting social bots by simultaneously modeling social behaviors and content. The study aimed to create accurate profiles of social users by simulating their social interactions and incorporating both internal and external factors. Previous studies had focused on extracting semantic information and content representations using temporal user information. However, this study was the first to employ Deep Learning (DL) techniques for identifying social bots through modeling social users. The effectiveness of the proposed approach was demonstrated through its application on real-world datasets.

A dataset consisting of manually coded instances of false news, satire tales, and corresponding rebuttal stories was introduced by researchers [5] . The dataset not only provides confirmed false news examples but also includes a thematic content analysis of these pieces. The analysis identified key themes such as exaggerated praise or denunciation of individuals, conspiracy theories, racial themes, and the dismissal of credible sources. The researchers made this dataset publicly available for research purposes and conducted evaluations on it. The results showcased promising language features for classification purposes.

[6] focused on identifying news transmission routes for the detection of fake news on social media. The study employed tuples representing user characteristics involved in the dissemination of news. The propagation paths of news items were modeled as multivariate time series data. Time series classifiers, combining recurrent and convolutional networks, were then used to capture global and local variations in these propagation paths, enabling the detection of fake news. Through experimentation on three real-world datasets, the study demonstrated the effectiveness of their approach, achieving accuracy rates of 85 %.

[7] put forth a Convolutional Neural Network (CNN) architecture based on user attention, coupled with adversarial cross-lingual learning. Their proposed architecture leveraged user attention mechanisms to capture the language-specific characteristics present in their posts. This attention- based CNN model was then integrated into an adversarial cross-lingual learning framework to eliminate language-specific and independent features, enhancing user posts. This approach effectively addressed data insufficiency challenges by utilizing user properties as language bridges. The suggested scheme showcased notable improvements compared to other techniques, particularly on English and Chinese microblog datasets.

[8] delved into the challenges surrounding unidentified characteristics of fake news and the interconnections between news articles, their creators, and subjects. In response, they proposed a system called "Fake Detector" that automates the process of determining the trustworthiness of fake news by conducting thorough data analyses. The Fake Detector system employed deep diffusive networks to simultaneously learn about news articles, their creators, and subjects, utilizing multiple characteristics extracted from both explicit and concealed content within the text. Extensive testing demonstrated the effectiveness of this schema, surpassing numerous other algorithms when applied to real-world datasets of false news.

[9] proposed the use of pattern-driven networks to detect fake news within social networks. Their approach considered the false information itself, the individuals responsible for spreading it, and the connections between them. Additionally, they provided explanations based on social psychology theories for the occurrence of such false news. These patterns were then leveraged to identify false news across various network levels, including individual nodes, egos, triads, communities, and the overall network. Through experiments conducted with real-world data, their approach demonstrated higher performance compared to most similar methods in terms of accuracy and effectiveness.
[10] Proposed that false news and disinformation can manipulate public opinions and influence political processes. Therefore, it is crucial to detect false statements to gain a better understanding of their dissemination and prevent their spread. However, existing techniques that categorize fabricated articles based on their content face significant challenges due to the ever-changing nature of news stories, particularly in response to new political events. This study addresses this issue by proposing a TAG (Topic-Agnostic) classification approach, which utilizes linguistic and web-markup elements to identify false news sites. The study presents experimental findings from multiple datasets, demonstrating that the proposed technique achieves higher accuracy in detecting false news even as the themes and topics evolve.

[11] put forward a theory-driven methodology for detecting false news. The proposed technique involves analysing news material at multiple levels, including lexicon, syntax, semantics, and discourse. By leveraging well-established concepts from forensic and social psychology, the approach examines news content comprehensively. The detection of false reports is then conducted using supervised machine learning frameworks. This interdisciplinary project investigates emerging trends in false news, improves the interpretability of false news feature engineering, and uncovers connections between clickbait, false news, and deception/disinformation. The suggested methodology outperformed most other techniques when evaluated on two real-world datasets containing limited information.

In their research [12] proposed the utilization of Bi-Directional Graph Convolution Networks (Bi-GCN) to analyze the characteristics of fake news. The study employed both top-down and bottom-up approaches, examining the propagation and diffusion of rumors. Specifically, the study employed Graph Convolution Networks (GCNs) with directed graphs representing the spread of rumors and with opposing graphs representing the diffusion of rumors. Additionally, the GCN layers incorporated information from the original posts to enhance the impact of rumors. Through empirical evidence, the study demonstrated the superior performance of their method compared to similar techniques.

In their research, [13] emphasized the importance of detecting false news by assessing its reliability. By analyzing publicly available data on inaccurate news, the study found that information about the news sources (writers) can serve as strong indicators of accuracy. The results indicated that the author's previous associations with false news can be utilized to identify it. This methodology has the potential to improve conventional approaches to detecting false news, which often relies on analyzing content characteristics to identify false statements.

In their research [14], the CNIRI-FS model was proposed. This model utilized the semantic properties of Wikipedia in conjunction with external enrichments derived from web page reliability to detect fake content. A genetic Algorithm (GA) was employed to eliminate untrustworthy attributes. Machine learning classifiers were then used to validate the best feature set, including features that handled negation. The CNIRI-FS model outperformed a comparison model in terms of precision and accuracy, as the latter did not utilize optimal feature selection.

In their study [15], a dataset of false news related to vaccines was utilized to evaluate various automatic algorithms for false news identification through statistical text analysis. It was observed that CNN (Convolutional Neural Network) demonstrated better performance in distinguishing be- tween large classes such as fake news and trusted news. On the other hand, the gradient-boosting decision tree (D.T.) with a feature stacking strategy showed better results for detecting satire. The study made contributions by showcasing that, despite the presence of larger classes, efficient identification of satire can be achieved through merged embeddings and a specialized model. Additionally, the study intentionally incorporated duplicate data to enhance the prediction of parody news from both false and reputable information.

In their research [16], a false news detection system was developed based on an advanced learning model. The initial preprocessing and editing of news reports were conducted, and several training models were evaluated. To identify false news, an ensemble learning model was employed, which combined four distinct models: LSTM, depth LSTM, LIWC CNN, and N-gram CNN. To enhance the accuracy of false news identification, the system utilized SAHS (Self-Adaptive Harmony Search) to determine the optimal weights for the ensemble learning model. The study demonstrated a maximum accuracy of 99.4%.

[17] addressed the identification of fraudulent information, including rumors, propaganda, or misleading content on social media platforms, as a significant problem with the potential to mislead users in their study. Specifically focusing on Twitter, a popular social networking platform, especially in the Arab world, the research aimed to develop an intelligent classification model for the early detection of incorrect information in Arabic tweets. The study employed NLP techniques, ML models, and HHO (Harris Hawks Optimizer) as a wrapper-based feature selector. A corpus of 1862 previously annotated Arabic-language tweets was used to evaluate the effectiveness of the model. The research utilized the BOW (Bag of Words) model with various term-weighting approaches for feature extraction. Multiple feature combinations were explored, including user profiles and content/word features, and eight well-known learning algorithms were examined. Among them, The combination of Logistic Regression (L.R) with Term Frequency-Inverse Document Frequency (TF-IDF) proved to deliver optimal outcomes in the context at hand. Moreover, the incorporation of binary Harris Hawks Optimizer (HHO) approaches for feature selection played a pivotal role in reducing dimensionality.
In their study, [18] discussed the rapid spread of the COVID-19 pandemic has been accompanied by a significant rise in disinformation and fake news related to COVID-19. This misinformation has confused people, emphasizing the need for an effective approach to detect and distinguish between genuine and misleading news regarding COVID-19. The study proposes an improved evolutionary detection strategy that aims to outperform previous approaches in detecting COVID-19 disinformation. The proposed method is designed to prioritize reducing the quantity of symmetrical features while upholding a high level of reliability. This objective is achieved through the integration of three wrapper feature selection techniques: Particle Swarm Optimization (PSO), Genetic Algorithm (GA), and the Slap Swarm Algorithm (SSA), which facilitate evolutionary classifications. The experiments carried out in the study are conducted on the Koirala database, a renowned dataset widely used in this particular domain. Based on the prediction outcomes, the proposed model exhibits a positive and exceptional performance in terms of predictability, achieving a high accuracy rate of 75.4% .Overall, the study contributes to the field of COVID-19 disinformation detection by presenting an improved evolutionary detection strategy that yields better results than previous approaches. The findings highlight the potential of the proposed method in enhancing the accuracy and reliability of detecting COVID-19-related disinformation.

[19] introduced a novel mode for false news detection that utilizes optimization approaches. The study proposes an enhanced Slap Swarm Algorithm (SSA) with a non-linear decreasing coefficient and oscillating inertia weight, alongside traditional SSA, Grey Wolf Optimization (GWO), and two adaptive SSA approaches. These optimization algorithms are evaluated on real-world datasets of fake news, demonstrating the superiority of the new model over SSA and GWO. By combining optimization techniques with false news detection, the research contributes to advancing the field and provides valuable insights into improving the accuracy of identifying false news.

[20] presents an innovative approach for detecting fake news using ant colony optimization algorithms with an adaptive objective function. The study focuses on improving the recall or precision of identifying fake or real news by analyzing the title of the news using natural language processing techniques. Additionally, a confined term matrix is utilized to limit the scope of the words analyzed, resulting in faster initial categorization. The research includes an analysis of a real dataset and provides classification results, specifically highlighting the recall and precision achieved through the adaptive objective function of the algorithm.
[21] introduces a novel approach for detecting fraudulent news articles by leveraging content- based features and the WOA-Xgbtree algorithm. This method offers the ability to classify news items across different contexts. The detection process involves two main steps: first, identifying and evaluating valuable characteristics from the news articles, and second, optimizing an Extreme Gradient Boosting Tree (xgbTree) algorithm using the Whale Optimization Algorithm (WOA) to categorize the articles based on the extracted features. The evaluation of classification accuracies is performed using F1 measures on a dataset containing approximately 40,000 news items. The proposed model is compared to various other classification techniques, demonstrating superior performance with higher F1 measure values achieved in 91%.
[22] aims to develop a reliable prediction model for detecting false news in Arabic Tweets by employing a combination of NLP techniques, feature selection methods, and advanced machine learning algorithms. The provided tweets are filtered and translated into a structured format using NLP methods. The Recursive Feature Elimination (RFE) approach is then applied to eliminate uninformative features. The prediction model is constructed using various machine-learning approaches. The experimental results underscore the superiority of the Logistic Regression (L.R.) classifier over alternative methods. Furthermore, the Recursive Feature Elimination (RFE) technique proves its efficacy in boosting the overall performance of the L.R. classifier. As a result, the proposed model attains a commendable prediction accuracy of 82%.
[23] presents a novel algorithm aimed at detecting COVID-19 disinformation and assisting scientists in differentiating between fake and accurate news. The proposed approach utilizes the Archimedes Optimization Algorithm (AOA) to improve classification results by reducing the number of features while maintaining high accuracy. Through the implementation of a wrapper feature selection technique using AOA, the algorithm demonstrates superior performance and accuracy in comparison to state-of-the-art classifiers and other nature-inspired evolutionary techniques, as shown by simulation results. This algorithm holds great promise in effectively identifying and combating COVID-19 disinformation, offering valuable support to the ongoing efforts in tackling misinformation during the pandemic.

The study presented in [24] introduces a hybrid model that capitalizes on the advantages of both Artificial Neural Network (ANN) and Auto-Regressive Integrated Moving Average (ARIMA) models. This hybrid approach effectively addresses the limitations of ANN models and yields more comprehensive and accurate forecasts in comparison to conventional hybrid ARIMA-ANN models. By combining the strengths of these two models, the proposed approach offers a promising solution for enhanced forecasting capabilities.. The approach leverages neural networks to capture and forecast underlying data, while also utilizing the distinctive properties of ARIMA models for linear modeling on preprocessed data. By effectively identifying and amplifying existing linear structures in the data, the proposed hybrid model demonstrates improved forecast accuracies compared to both traditional hybrid and singular models. The study's results, obtained from real- world datasets, validate the effectiveness of this approach in enhancing forecast accuracy.

[25]propose the use of polarity analysis for developing a false news detection model. They gather editorial items from prominent English-language news websites and carefully annotate them. The content-based features are constructed using n-grams, and these features are then processed by SVM, KNN, and Naive Bayes classifiers. The results indicate that KNN classifiers outperform SVM classifiers in polarity analysis for detecting fake news. This suggests that the KNN classifier is more effective in capturing the polarity patterns and distinguishing between true and false news articles.

[26] presents methods based on Machine Learning (ML) and Natural Language Processing (NLP) for gathering news and utilizing SVM to assess the legitimacy of articles posted on social media platforms such as Facebook, Twitter, WhatsApp, and microblogs. The developed model is compared to existing models, and the results demonstrate its effectiveness. The model achieves a high level of accuracy, accurately determining the correctness of news articles with up to 93.6 [62]incorporated content/context level information into tree-based ensembles, specifically Gra- dient Boosting, for the identification of false news. Adaptive boosts were developed using gradient descents to maximize standard goal functions by selecting key components and characteristics of the techniques. The study utilized multiclass datasets, specifically the False News Challenges (FNC), and multiple machine learning models. Experimental results demonstrated that their en- semble frameworks outperformed previous benchmarks. By employing the Gradient Boosting approach, which is an ensemble machine learning framework, the work achieved high accuracy in classifying false news into four categories, with accuracy reaching up to 86%.

[27] focused on detecting false news in Indonesian language texts by utilizing an ensemble learning strategy. They specifically addressed the challenge of unbalanced data in the provided dataset. The study evaluated over 660 documents and compared the performance of Random Forest (RF) ensembles with multinomial Naive Bayes (N.B.) and Support Vector Machine (SVM) classifiers (non-ensembles). The findings revealed that the RF ensembles achieved an impressive f1-score of 0.98, outperforming the non-ensemble classifiers, which scored only 0.43 and 0.74 for multinomial N.B. and SVM, respectively. Furthermore, when comparing their results to other studies using the same dataset, the ensemble approach demonstrated superior performance. These results highlight the effectiveness of the ensemble learning strategy, specifically the Random Forest model, in distinguishing between true and false news in Indonesian texts.

[28] discussed detection of false news in web publications using semantic characteristics and various machine-learning approaches. The study employed the Naive Bayes (NB) and Random Forest (RF) classifiers and evaluated their performance using five categories of linguistic variables. The results showed that utilizing bigram features and the RF classifier yielded the highest- performing model, achieving an impressive accuracy of 95.66%.

In reference to [29], the study concentrates on detecting false news associated with Covid-19 on social networks through the utilization of text data mining techniques and supervised machine learning algorithms. The research encompasses text mining analysis and classification procedures conducted on a designated dataset. To identify Covid-19 false news, two algorithms, namely Random Forest (RF) and Decision Tree (DT), were employed. Additionally, a modified version of the DT model was utilized as part of the detection process.. The experimental results showed significant improvements in terms of accuracies, precisions, recall, F1 measures, and support values when comparing the two algorithms. The RF algorithm achieved a trained and tested accuracy of 94.49%.

In reference [30], the focus is on detecting Bangla fake news from social media, highlighting the importance of attention in this area. The study employs two supervised machine learning algorithms, namely Multinomial Naive Bayes (MNB) and Support Vector Machine (SVM) classifiers, for the identification of Bangla false news. Feature extraction is performed using Count Vectorizer and TF-IDF vectorizer. The framework takes into account the polarity of the related article to recognize fake news. The results of the research show that SVM with a linear kernel outperforms MNB, achieving an accuracy of 96.64%. This study contributes to the understanding of detecting fake news in the context of the Bangla language and emphasizes the effectiveness of SVM in this task.

In their study, [31] researchers addressed the challenges of detecting fake news by leveraging deep learning (DL) architectures. The exponential production and dissemination of distorted news articles necessitate automatic identification and detection methods. However, recognizing the nuances of natural language poses a significant challenge for automatic fake news detection. Moreover, existing models often rely on binary classifications, limiting their ability to capture the closeness or distance between fake and genuine news. To overcome these limitations, the study employed neural networks and DL architectures to accurately predict attitudes based on combinations of news titles and article bodies. This approach surpassed previous models by 2.5% and achieved a test data accuracy of 94.21%. By considering the relationship between titles and content, the models demonstrated their effectiveness in improving the accuracy of fake news detection.

In the study conducted by [32] , a comparison was made between classification results obtained from expert and amateur annotations to understand how annotator knowledge of hate speech influences classification models. The research focused on hate speech annotations for 6,909 tweets, which were annotated by two groups: CrowdFlower annotators (referred to as amateur annotators) and annotators with theoretical and applied knowledge of hate speech (referred to as expert annotators). The findings revealed that expert annotations could generate models that performed similarly to previous classification efforts. The study suggests further investigation into socio-linguistic characteristics such as gender and position to gain deeper insights into hate speech classification.

In a study discussed in reference [33] , the focus was on automated hate speech detection and the challenge of identifying offensive language. Hate speech was defined as language expressing hatred towards a specific group or intending to be negative, demeaning, or insulting towards its members. The researchers collected tweets containing hate speech keywords using a crowd-sourced hate speech lexicon. A subset of these tweets was categorized into three groups through crowd-sourcing: those containing hate speech, offensive words, and none. To differentiate between these groups, a multi-class classifier was trained. The highest-performing model achieved an overall precision of 0.91, recall of 0.90, and F1 score of 0.90. However, around 40% of hate speech instances were misclassified, resulting in an accuracy score of 0.44 and a recall score of 0.61 for the hate speech class. Additionally, approximately 5% of offensive tweets and 2% of harmless tweets were wrongly labeled as hate speech. The authors emphasize the importance of examining individuals who engage in hate speech, including their characteristics, motivations, and the social systems contributing to such behavior. They also highlight the significance of recognizing and addressing societal biases that may impact the performance of hate speech detection algorithms.

3. Research Methodology

Social networking platforms (OSNs) have become crucial for communication and information dissemination. To tackle the problem of spread of misinformation on social networks, we proposed a methodology, as depicted in Figure 1. The methodology consists of several stages: Data Extraction, Data Preprocessing, Feature Extraction, and Deep Learning Classification.

Based on this methodology, a model named Enhanced Feature Engineering For Fake News Identification (EFEFI) has been developed. EFEFI aims to effectively identify and classify propaganda content on social networks, thereby aiding in the reduction of misinformation and the promotion of more accurate and reliable information dissemination. We also used multiple epochs and set conditions to terminate at the point when the improvement stops to avoid issue of overfitting. In the upcoming sections we will explore the different stages of this methodology.

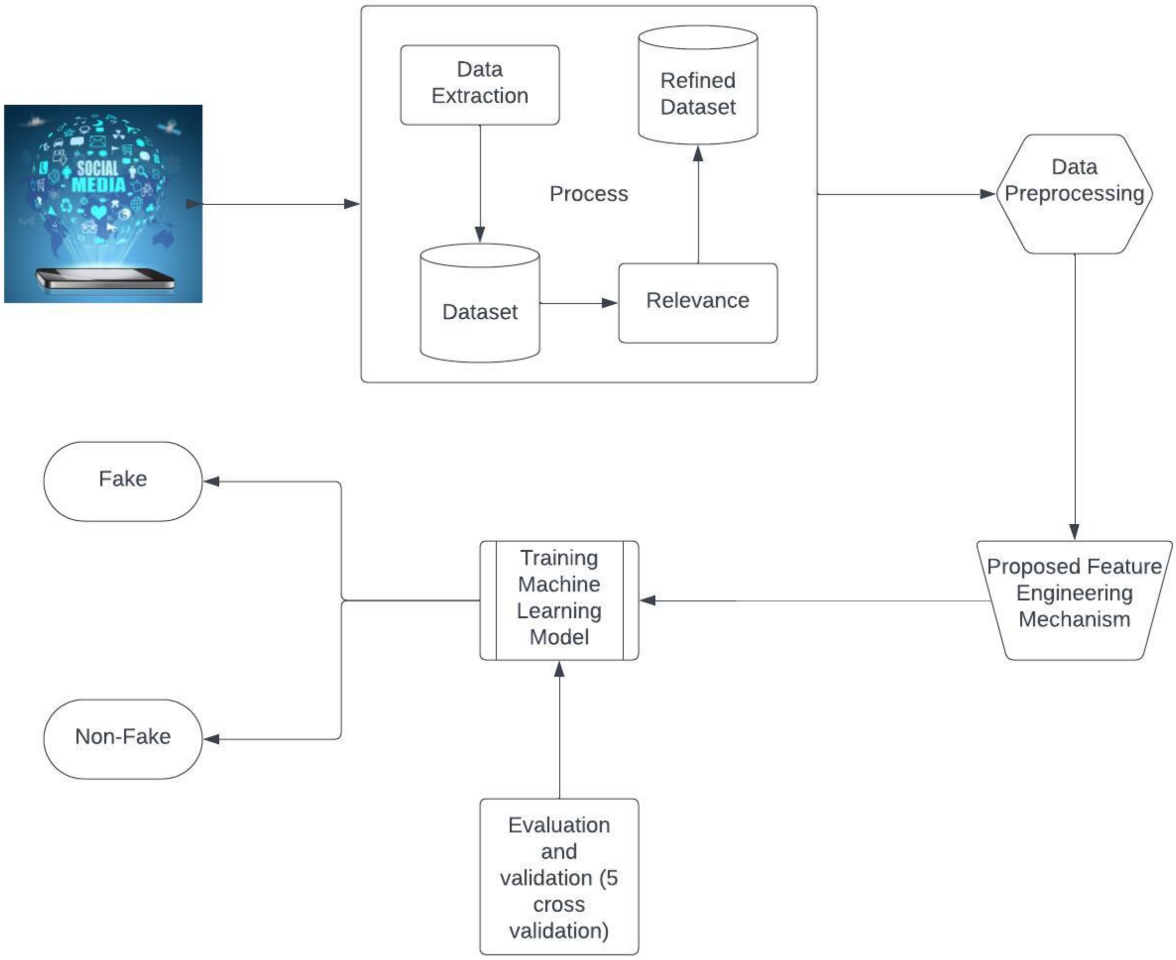


Figure 3. 1 - Methodology adopted for Classification of Fake and Non-Fake News

3.1 Data Collection

Data collection is a critical aspect of research, and in this study, we utilized data extracted from Twitter, a popular social networking platform available over internet. We used 'Pheme dataset" and then also run our model on other datasets like LIAR, FakeNewsNet, and BuzzFeedNews.

Twitter, as a dynamic and interactive micro-blogging platform, serves as a vibrant space for users to share short updates known as tweets. These concise messages, limited to 280 characters, appear on users' timelines, creating a continuous stream of information and conversations. Within these tweets, users often refer to various entities, such as people, organizations, and events, providing a glimpse into real-world occurrences and discussions. With its massive user base and constant activity, Twitter generates an immense volume of tweets every day, making it a valuable resource for academic research and commercial applications alike. To access and retrieve Twitter data, developers use the Twitter API, which offers three main types: Search API, Streaming API, and REST API.

The Twitter API provides various methods for developers to access and retrieve data according to their specific requirements. The Search API allows users to query tweets based on keywords, mentions of individuals, or specific user tweets. While it aims to provide relevant results, there may be cases where some data is missing. On the other hand, the Streaming API offers real-time data by establishing a continuous HTTP connection, enabling the collection of a large volume of real-time data based on keywords or location-based parameters. However, it's important to note that the Streaming API provides only a sample of the real-time stream. Lastly, the REST API grants access to past data of individual users, including timelines, status updates, and user details. It also provides functionality for users to interact with the platform. Each API serves a specific purpose and offers different capabilities for data retrieval and interaction, allowing researchers and developers to choose the most suitable option based on their project requirements. The framework for data extraction is shown in Figure 3.2.

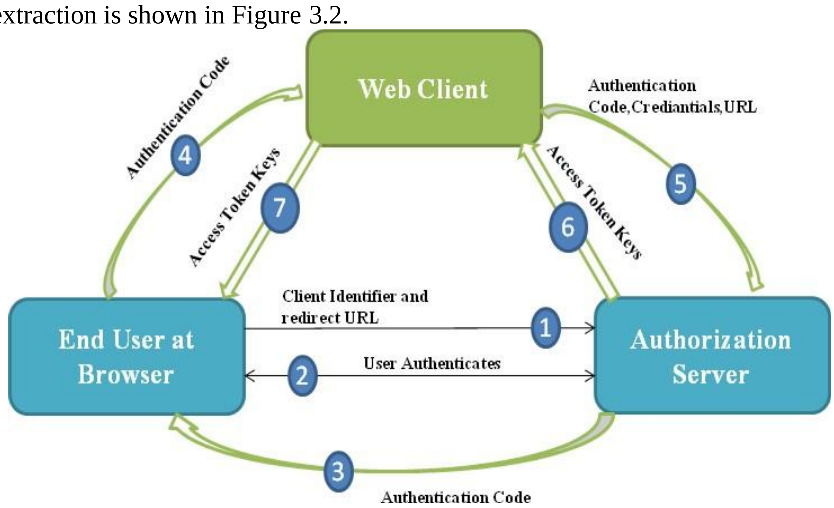


Figure 3. 2 - Framework for Data Extraction from Twitter.

3.2 - Data Preprocessing

Data extracted from online social networks often contains noise, which cannot be directly used by machine learning algorithms for training and testing. Hence, preprocessing is crucial to refine the noisy data. In classification applications, preprocessing plays a significant role in understanding the meaning of short texts. Although it has received less attention in the literature compared to feature extraction and classification, it greatly impacts the overall system performance. Preprocessing of tweets involves preparing them for various tasks such as event recognition, fake news detection, sentiment analysis, and more.

On social media platforms like Twitter, individuals tend to follow their own informal language rules. As a result, each Twitter user exhibits their unique writing style, including the use of abbreviations, non-standard punctuation, and misspelled words. Emoticons and emojis are commonly employed in tweets to convey complexity, sentiment, and ideas. Slang terms, acronyms, URLs, hashtags, and user mentions are also prevalent in tweets.

3.3 - Data Collection:

Collecting data is a crucial part of any research, and in our study, we got our data from Twitter, a popular social media platform. We started with the 'Pheme dataset' and also looked at other datasets like LIAR, FakeNewsNet, and BuzzFeedNews.

Twitter is known for its short updates called tweets, which users share on their timelines. These tweets are limited to 280 characters, so they're brief but can convey a lot of information. People on Twitter talk about all sorts of things, like individuals, organizations, and events, which gives us a glimpse into what's happening in the real world.

Twitter is always buzzing with activity, with millions of tweets posted every day. That makes it really useful for both research and business purposes. To get data from Twitter, developers use something called the Twitter API, which comes in different types like Search API, Streaming API, and REST API. The word cloud of data collected initially is as in Figure 3.3.

Figure 3.3 : Word Cloud

3.4 - Data Preprocessing

Data gathered from social platforms often contains noise, making it unsuitable for direct use by deep learning algorithms in training and testing. Therefore, preprocessing is essential to clean and refine this noisy data. In classification tasks, preprocessing has an important role in interpreting of short texts. Although it has garnered less focus in research compared to extraction and classification process, it plays a crucial role in influencing the overall performance of the system. Preprocessing tweets is crucial for various tasks, including event detection, identifying fake news, sentiment analysis, and other similar applications. Some of the preprocessing steps include:

➢ Data Augmentation:

Data Augmentation is an important process in Deep learning and analysis of data, involving the adjustment of class distribution within a dataset to ensure fair representation of different classes or categories. Imbalanced data occurs when one class dominates the dataset, while others are poorly represented . Dealing with imbalanced data is crucial as it can lead to biased models that struggle to accurately predict the minority class.

SMOTE (Synthetic Minority Over-sampling Technique) is a highly effective approach considered to tackle the issue of class imbalance in datasets. It primarily aims to tackle the issue of minority class underrepresentation by creating synthetic samples that closely match the existing instances. The technique works by selecting instances from the minority class and generating synthetic examples through interpolation of feature values based on the nearest neighbors. This approach helps to improve the representation of the minority class, resulting in a more balanced class distribution in the dataset. By using SMOTE, classifiers can learn from a broader set of diverse examples, reducing bias towards the majority class and ultimately improving model performance. This technique is especially useful in classification tasks where the minority class is underrepresented, offering an effective solution for addressing class imbalance issues.

➢ Tokenization:

Tokenization is an essential stage in the processing of text for natural language processing (NLP). It entails dividing raw text into smaller units called tokens, which may include words, phrases, or symbols. involves splitting raw text into smaller units known as tokens, which can be words, phrases, or symbols.

➢ Normalization:

Normalization ensures that data features are scaled similarly, preventing certain features from dominating others during the model training process. Few common normalization techniques are Min-Max Scalling, Z-Score Normalization and Robust Scaling.

➢ Removal of Noise, URLs, Hashtags, and User Mentions:

Social media text often contains noise in the form of URLs, hashtags, and user mentions, which can distract from the underlying message. Let's delve into how we clean up the text:

- URL Removal:

URLs provide links to external content but are unnecessary for text analysis. Removing URLs simplifies the text and prevents them from being treated as unique tokens by the model.

- Hashtag Removal:

Hashtags serve to categorize or label content but do not add to the semantic meaning of the text. Removing them helps simplify the text and allows the model to focus on the more meaningful elements.

- User Mention Removal:

User mentions, indicated by "@" symbols followed by usernames, are references to specific individuals or entities. While relevant for social interactions, they are typically irrelevant for text analysis tasks and can be safely removed.

- Noise Removal:

Noise encompasses any irrelevant or extraneous information present in the text, such as special characters, emojis, or non-standard punctuation. Removing noise helps clean up the text and improves the model's ability to extract meaningful features.

➢ Word Segmentation:

Word segmentation is essential for languages in which words are not clearly distinguished by spaces or punctuation marks. . Let's explore how we tackle word segmentation:

- Rule-Based Segmentation: Rule-based segmentation relies on predefined rules or patterns to identify word boundaries. This approach is straightforward but may struggle with languages that exhibit complex morphology or lack clear delimiters between words.

- Statistical Segmentation: Statistical segmentation leverages probabilistic models to identify word boundaries based on statistical patterns in the text. This approach is more flexible and adaptable to different languages and writing styles.

- Machine Learning-Based Segmentation: Machine learning algorithms, such as recurrent neural networks (RNNs), can learn to segment words from raw text based on annotated training data. These models can capture subtle linguistic cues and nuances, making them well-suited for diverse text segmentation tasks.

➢ Replacing Emoticons and Emojis:

Emoticons and emojis add emotional context and expressiveness to text but can pose challenges for text analysis algorithms. Here's how we handle them:

- Emoticon Replacement: Emoticons, such as ":)", ":(", or ":D", are converted into their corresponding textual representations (e.g., "smile", "sad", "laugh") to standardize the text and ensure consistency across different platforms.

- Emoji Replacement: Emojis, graphical icons representing emotions, objects, or concepts, are replaced with descriptive terms that capture their intended meaning.

By replacing emoticons and emojis with textual equivalents, we ensure that the sentiment and meaning conveyed by these symbols are preserved in the text analysis process.

➢ Abbreviations and Slang:

Abbreviations and slang are prevalent in informal communication, such as social media posts and text messages. Here's how we handle them:

- Abbreviation Expansion: Abbreviated phrases, acronyms, and shorthand expressions are expanded into their full forms to facilitate understanding and interpretation. For example, "LOL" might be expanded to "laugh out loud", and "BRB" to "be right back".

- Slang Translation: Slang terms and colloquial expressions are translated into their standard equivalents to ensure consistency and clarity in the text. For instance, "gonna" might be replaced with "going to", and "gr8" with "great".

By expanding abbreviations and translating slang, we make the text more accessible and interpretable for automated analysis.

➢ Punctuation Removal:

Punctuation marks convey grammatical structure and emotional tone in text but can interfere with automated text processing tasks. Here's how we handle punctuation:

- Punctuation Removal: Extraneous punctuation marks are removed from the text to simplify its structure and improve readability. This ensures that punctuation does not interfere with subsequent text analysis steps, such as tokenization or part-of-speech tagging.

- Preserving Contextual Punctuation: Certain punctuation marks may carry valuable semantic information and emotional cues. In such cases, these marks are retained or replaced with appropriate tags to preserve their contextual meaning.

By removing unnecessary punctuation, we streamline the text and facilitate downstream text processing tasks without sacrificing important linguistic information.

➢ Stopword Removal:

Stopwords are common words that appear frequently in text but have little semantic value, such as "the," "is," and "and." Here's how we typically handle them:

Stopword Removal: Stopwords are eliminated from the text to minimize irrelevant noise and highlight content-bearing words that hold meaningful information. This step improves the efficiency and accuracy of text analysis tasks, such as keyword extraction or sentiment analysis.

Contextual Stopword Handling: In some contexts, stopwords may have significance. In these cases, stopwords are either retained or treated differently to preserve important linguistic nuances.

By eliminating stopwords, we improve the signal-to-noise ratio in the text, allowing for more accurate and insightful text analysis.

➢ Stemming and Lemmatization:

Stemming and lemmatization are approaches used to transform words into their fundamental or base forms, simplifying text processing and analysis. Here's how they work:

- **Stemming**: Stemming removes prefixes, suffixes, and inflectional endings from words to obtain their base forms, known as stems. This process simplifies text by reducing words to their core components, making text matching and retrieval more efficient. For instance, the words "running," "runs," and "ran" can be reduced to the base form "run."
- **Lemmatization**: Lemmatization goes a step further than stemming by mapping words to their canonical forms, known as lemmas, based on their dictionary definitions. This approach ensures that words are transformed into their most basic and meaningful representations. For instance, "better", "best", and "good" would all be lemmatized to "good".

By applying stemming and lemmatization, we standardize word forms and reduce the dimensionality of the vocabulary, leading to more robust and accurate text analysis results.

These preprocessing techniques collectively enable us to clean, standardize, and transform raw text data into structured representations that are suitable for Deep learning and text analysis tasks. By preparing the data in this manner, we can extract valuable insights, detect patterns, and make informed decisions based on textual information.

3.5 - Feature Engineering

Feature extraction is an essential phase in any classification task, as it focuses on identifying and extracting key attributes or qualities from the data that are important for deep learning models.

In context of text classification, feature extraction focuses on selecting and representing the key elements of the text that capture its essence and contribute to the classification process. For instance, when classifying tweets based on the expressed ideology, we aim to extract features that reflect extremist or soft ideologies. This process entails utilizing various techniques, such as statistical methods, information gain analysis, or domain-specific knowledge, to pinpoint the most relevant and distinguishing features. By selecting these meaningful features, we enhance performance of the Deep learning models in classifying and understanding the underlying patterns of the data.

- TF/IDF -

TF-IDF is a method used in natural language processing essential for extracting meaningful features and effectively representing text data. It works by combining two key elements: the occurrence of a term within a document (TF) and its rarity across the whole corpus (IDF) to evaluate the term's significance. While TF measures how often a term appears in a document, IDF evaluates its significance by considering how frequently the term appears across all documents in the corpus. By balancing these two elements, TF-IDF assigns greater importance to terms that appear frequently in a document but are uncommon in the overall corpus, making them more informative and relevant. This approach helps capture unique document characteristics, aiding tasks like document classification, information retrieval, and keyword extraction with greater accuracy.

- Bag of Words

The Bag-of-Words technique serves as a cornerstone in text feature extraction for Deep learning applications. By representing text solely based on word occurrence within a document, it disregards the inherent order or structure of the text. In essence, each document is treated as a "bag" filled with words, with their frequencies serving as the features. This method supports effective processing and analysis of text data by converting it into a numerical format.The approach hinges on two fundamental components: a vocabulary comprising known words and a metric for assessing their presence. Aptly named a "bag" of words, this technique discards any information pertaining to word order or structure within the document. Instead, it focuses solely on word occurrence, indifferent to their positional context. This simplicity and adaptability render the approach highly versatile, offering a plethora of applications in feature extraction from documents. It encompasses various forms, including unigram, bigram, and trigram words and lemmas, thereby enriching the feature space for Deep learning classifiers and enabling the extraction of comprehensive information from textual data.

Sentiment Analysis

In this stage, sentiments are derived from tweets by assigning values to individual words. Tweets are then categorized as positive, negative, or neutral based on these scores. As this is a classification task, a lexicon-driven approach is employed for sentiment analysis. The results showed that tweets with Fake content predominantly express neutral sentiments, whereas non-fake tweets tend to feature a mix of both positive and negative sentiments. This sentiment information is subsequently used for training and evaluation of model, enhancing its ability to accurately classify and categorize tweets.

- News Length

Following exploratory analysis, a notable finding emerged: sentences containing fake news tend to be longer compared to those conveying non-fake information. For categorization purposes, a tweet with fewer than eight tokens is considered a short document, while tweets with more than eight tokens are classified as long documents. Discrete features like Text Length and count of words are incorporated into the classification task for identification of fake news.

3.6 - Pseudocode 1 EFEFI: Hybrid Feature Engineering Approach for Fake News Identification

*1: **Require**: News (Pinput.csv),*
Classifier_Name, Classifier_Hyperparameters
*2: **Data Augmentation:** Pinput.csv (SMOTE)*
*2**: Ensure**: Fake News(FN) and Non-Fake News (NFN)*
*3: **for a** from **1** to **n** (Total No. of news items) **do***
4: C [a] = Pinput[a] $ Label // Addition of labels
5: posts.csv = C [a] //Data file containing the posts with labels

*6: **end for***
7: Pre = tokens (posts.csv) // tokenization
8: Pre = tokens_lower (Pre) //Changing to lower case letters.
9: Pre = tokens_remove (Pre) // Removing stop words.
10: Pre = tokens_stem (Pre) // Reducing a word to its base or root form
11: posts2.csv = Pre // Processed csv file
*12**: for a** from **1** to **n do***
13: P_length[a] = nchar posts2[a]) // News Length
*14: **end for***
15: Tokens = tokens_ngrams (posts2, n= 1:3)
// processed to extract n-gram features, specifically up to 3-grams
16: Tokens.dfm = dfm (Tokens) //create document feature matrix
17: dfm_trimm = dfm_trim (Tokens.dfm, min_docfreq, min_termfreq)
// Removing the tokens with low importance.
18: TFIDF_features = dfm_trimm. Tfidf // Retrieving TF/IDF Features
19: optimal_Features=TFIDF_features + P_length[a].
*20. **Algo** (Algo_Name, Algo_ Hyperparameters, optimal _Features)*

Word cloud after the algorithm is run is depicted below in Figure 3.4.

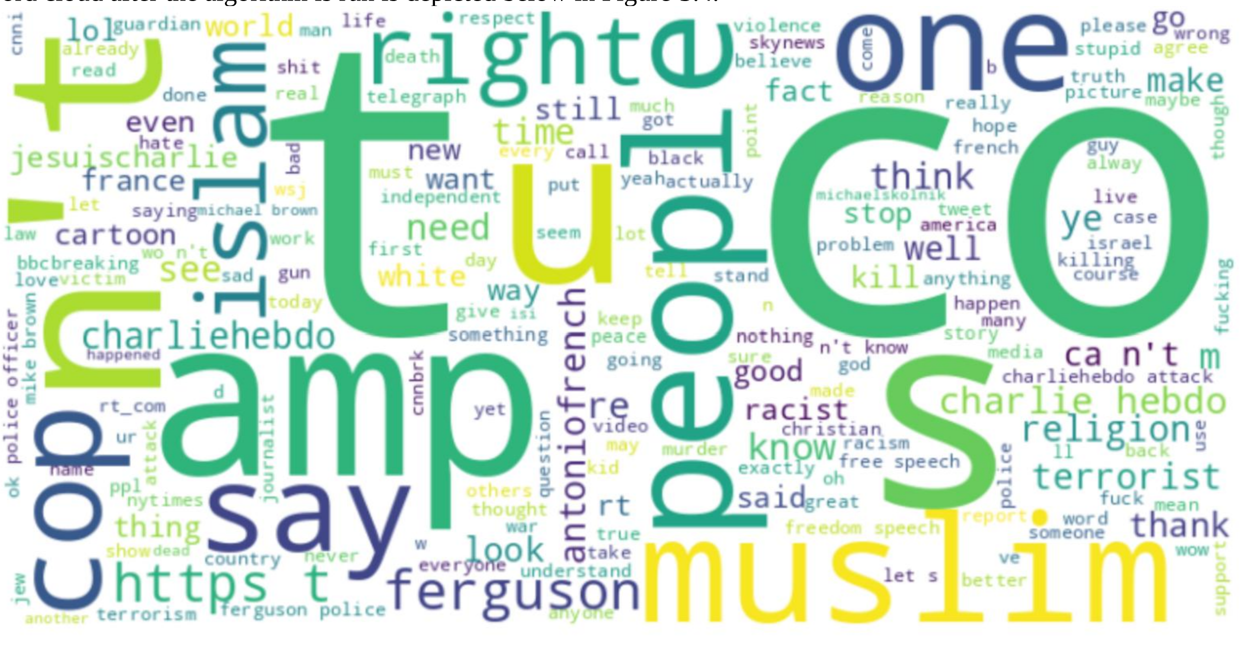


Figure 3.4 – Word Cloud After EFEFI

3.7 - Machine Learning Models Used in the Research –

3.7.1 - Logistic Regression Based EFEFI.

Logistic Regression is a widely used supervised machine learning algorithm that is particularly well-suited for binary classification tasks. In our study, we employed Binomial Logistic Regression, which is specifically designed for cases where the target variable can take on only two possible values: Fake and Non-Fake. The objective of Logistic Regression is to estimate the probability of an instance belonging to a particular class based on its features. Unlike other algorithms that directly predict class labels, Logistic Regression calculates a numerical value known as the log-odds or logit. This logit represents the likelihood of an instance belonging to the positive class (Fake) relative to the negative class (Non-Fake).

In our implementation of EFEFI (LR-EFEFI), we utilized the logistic regression algorithm to classify tweets as Fake and Non-Fake. By training the logistic regression model on the annotated dataset and fine-tuning the parameters, we obtained a model capable of predicting the class labels for the data. After applying EFEFI the ROC curve generated is shown as in Fig. 3.5.

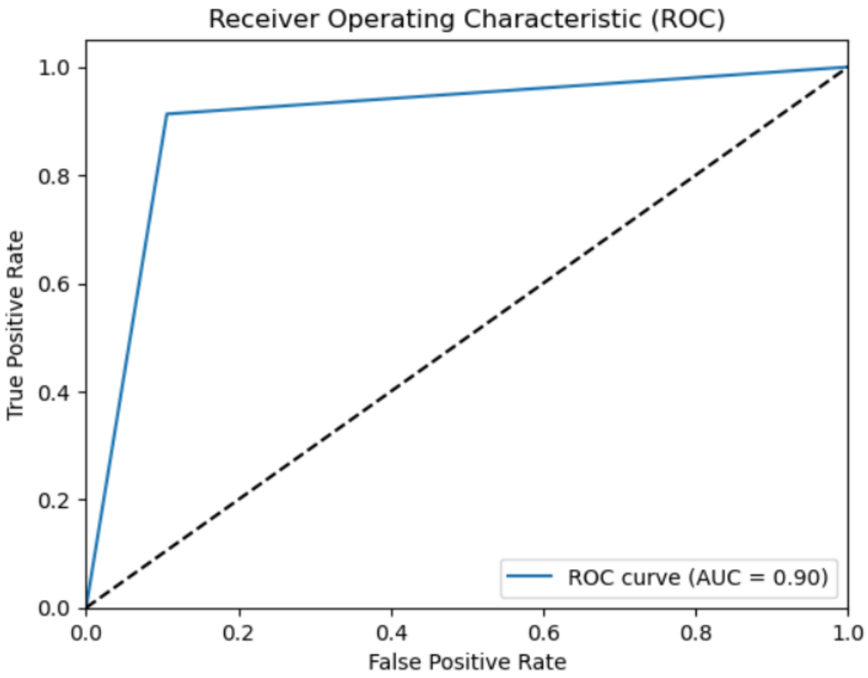


Figure 3. 5 - ROC Curve LR-EFEFI

3.7.2 - Support Vector Machine Based EFEFI

The Support Vector Machine (SVM) algorithm is a powerful tool in the field of supervised machine learning, particularly for text classification tasks. Its core principle revolves around finding an optimal hyperplane that effectively separates different groups of people. The key players in this process are the "Support Vectors," which are data points located in close proximity to the hyperplane. Altering or removing these support vectors can lead to a significant shift in the position of the hyperplane. The margin, on the other hand, refers to the gap between the support vectors and the hyperplane. In SVM, a specific text's features are taken into consideration to determine its label. In our case we implement SVM-based EFEFI (SVM-EFEFI) approach. Upon executing the pseudocode, the resulting ROC curve is displayed in Figure 3.6, providing insights into the classification performance.

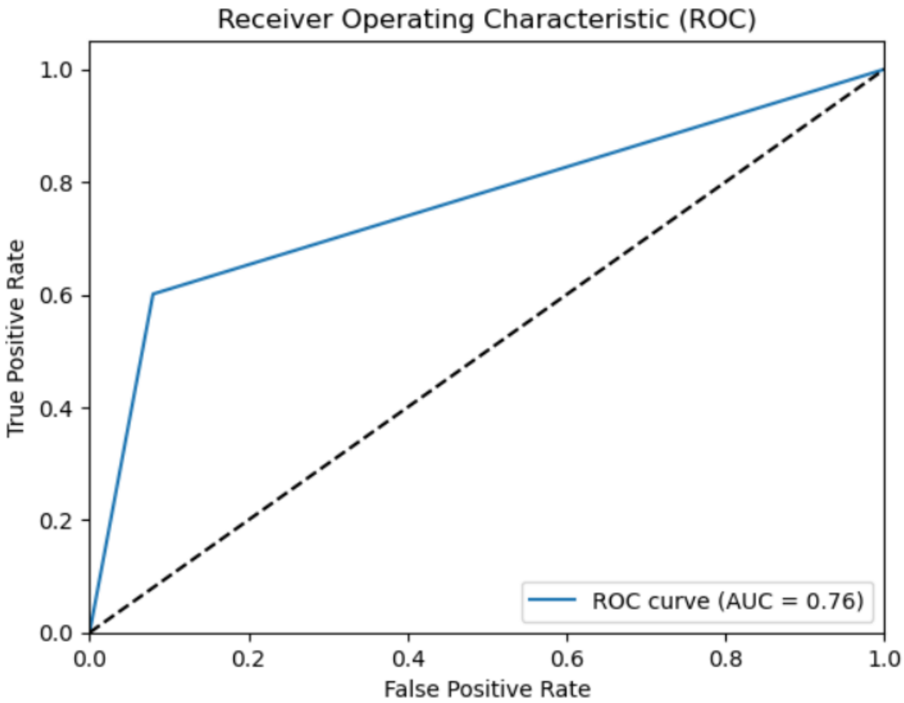


Figure 3. 6 – ROC Curve SVM-EFEFI

3.7.3 Decision Tree Based EFEFI

Decision trees are powerful data structures that resemble the shape of an actual tree. To construct a decision tree, we need to gather training data, which will serve as the foundation for generating predictions on test data. The underlying principle of decision trees is to achieve the highest level of accuracy with the fewest number of decision steps. This approach is well-suited for both classification and regression problems. By dividing the input space into distinct regions and assigning different classifications to each region, decision trees can effectively tackle complex tasks. In our case, after implementing the provided pseudocode, we obtain ROC curve, as illustrated in Figure 3.7, which provides valuable insights into the performance of the decision tree classifier.

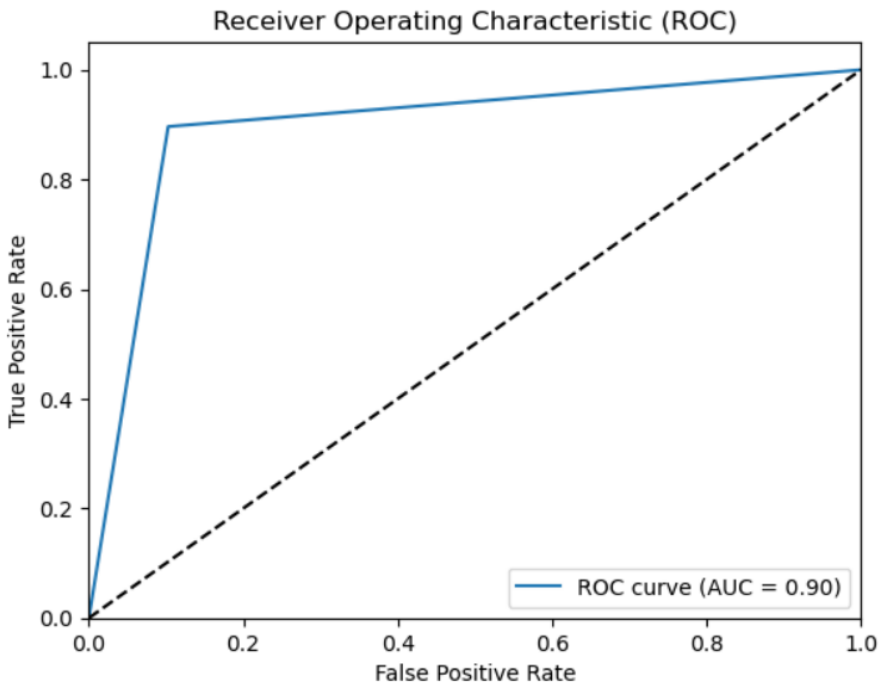


Figure 3.7 – ROC Curve DT-EFEFI

3.7.4 - Random Forest Based EFEFI

Random Forest is a widely recognized ensemble learning algorithm that combines multiple classifiers to deliver optimal solutions for complex problems. It leverages the power of multiple decision trees to make accurate predictions. By training a collection of decision trees on different subsets of the data and combining their outputs, Random Forest enhances the overall performance and robustness of the classifier. In our case, upon implementing the proposed pseudocode, we obtain a confusion matrix that reveals important insights into the classification results. The resulting ROC curve, depicted in Figure 4.6, provides a comprehensive overview of the Random Forest's performance in categorizing the data accurately.

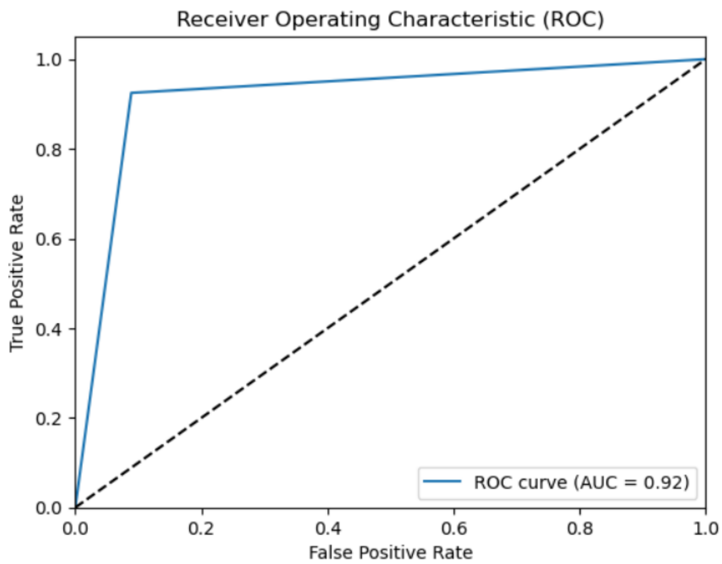


Figure 3.8 – ROC Curve RF-EFEFI

3.7.5 Voting Ensemble Method Based EFEFI

Voting is a powerful ensemble machine learning model that harnesses the collective wisdom of multiple individual models to enhance overall prediction performance. It serves as a technique to improve upon the performance of any single model within the ensemble. The concept behind a voting ensemble revolves around combining the predictions generated by diverse models. It is applicable to both classification and regression tasks. For regression, the ensemble calculates the average of the predictions from the individual models. After implementing our Pseudocode, we get the ROC curve as in Fig. 4.7.

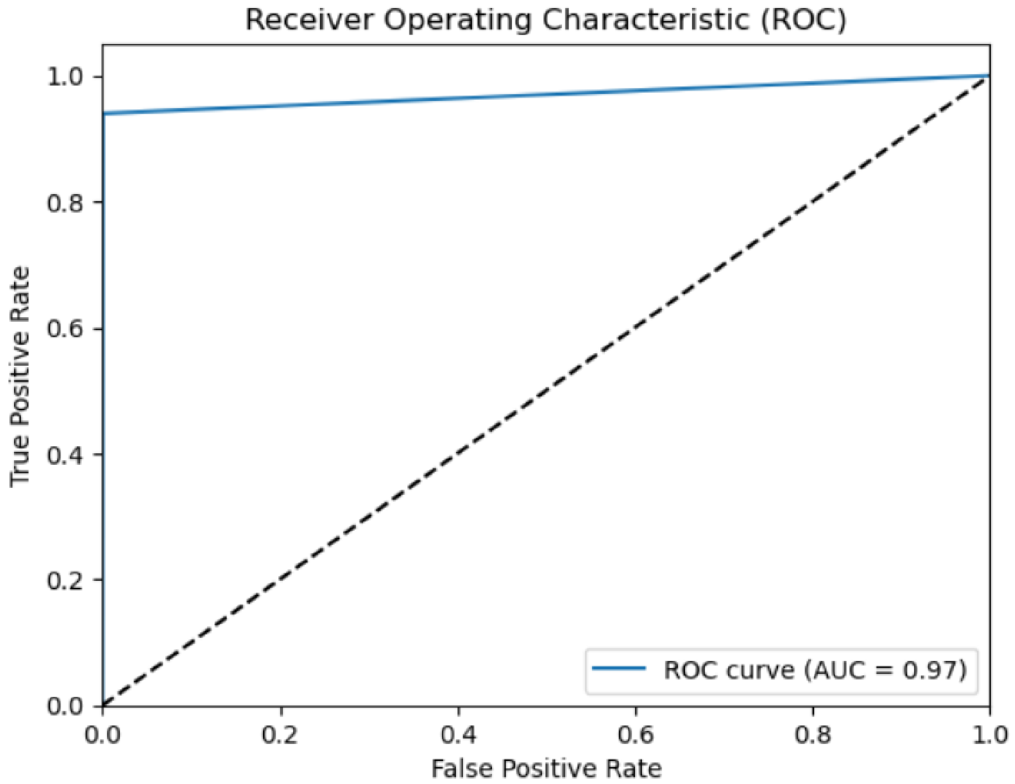


Figure 3.9 – ROC Curve VE-EFEFI

4. Results and Analysis -

In this section, we will delve into the outcomes presented in chapter 4 and examine the performance of the different algorithms. We will compare their performance both before and after implementing our model, EFEFI. Additionally, we will assess the performance of the algorithms following the implementation of EFEFI and ultimately select our final model based on these evaluations. Accuracy serves as a crucial criterion in this research for evaluating the effectiveness of the proposed framework. The metrics obtained from the confusion matrix are compared with other classification performance measures to showcase the performance of the proposed model. Precision, recall, f1-score, and accuracy are derived from the confusion matrix, providing valuable insights into the model's performance.

4.1.1 Logistic Regression

Confusion Matrix using Logistic Regression Algorithm before and after using EFEFI is depicted in figure 4.1 and 4.2 below:

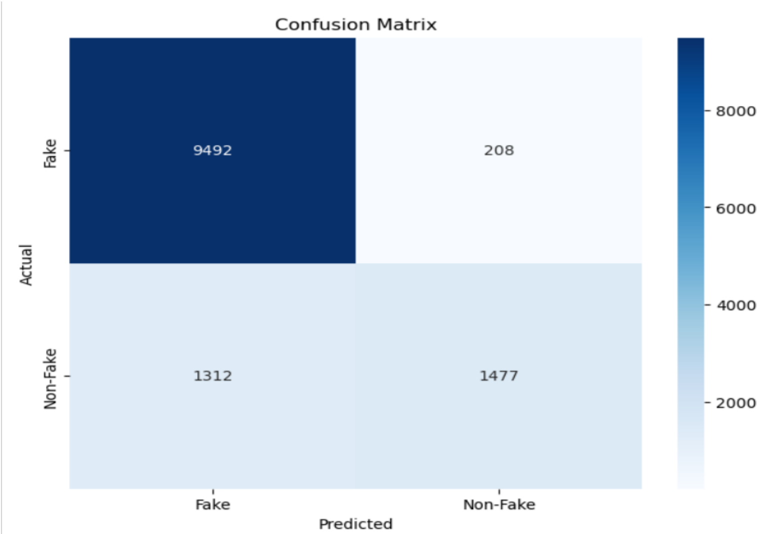


Figure 4. 1 – Confusion Matrix Before EFEFI – LR

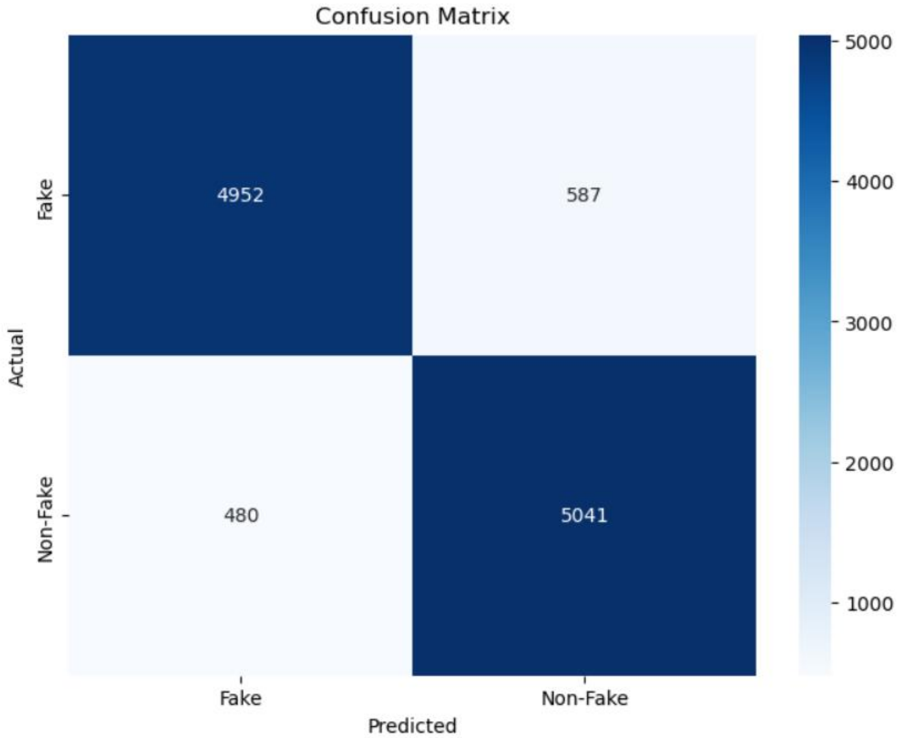
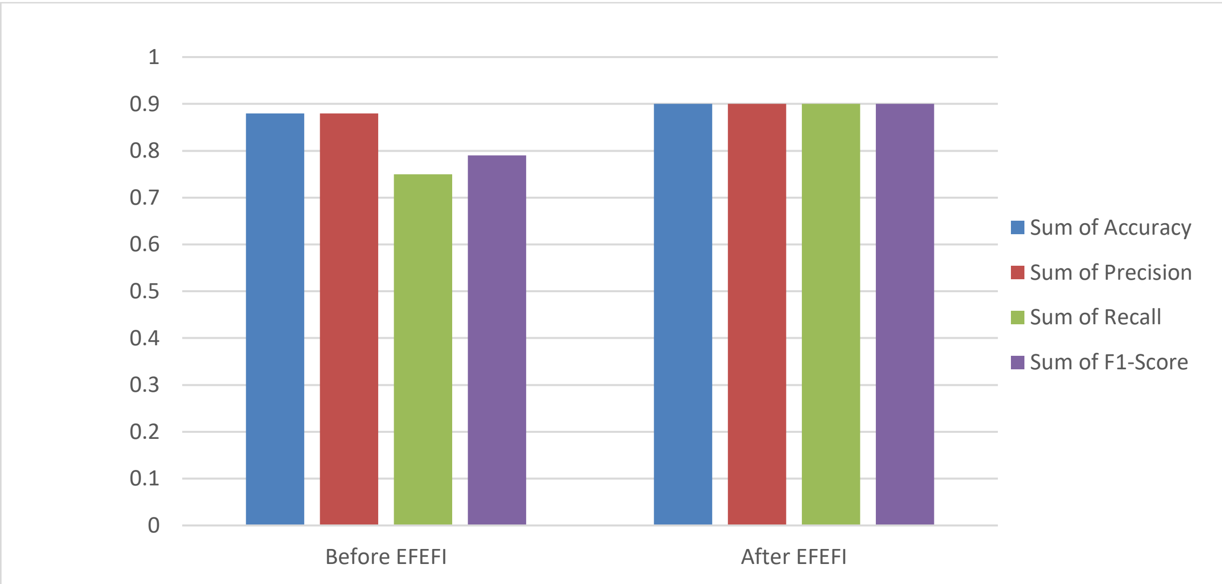


Figure 4. 2 Confusion Matrix After EFEFI – LR

Based on the confusion matrix above we find accuracy, Precision, F1 score and recall values before and after EFEFI and the same is shown in the Table 4.1 and depicted in graphical form in Graph 4.1.

LR Algorithm	Accuracy	Precision	Recall	f1-Score
Before EFEFI	88 %	88 %	75 %	79 %
After EFEFI	90 %	90 %	90 %	90 %

Table 4. 1 – LR Comparison Before and After EFEFI



Graph 4. 1 – LR Comparison Before and After EFEFI

4.1.2 Support Vector Machine

The Confusion Matrix for the Support Vector Machine, both before and after applying EFEFI, is illustrated in Figure 4.3 and Figure 4.4.

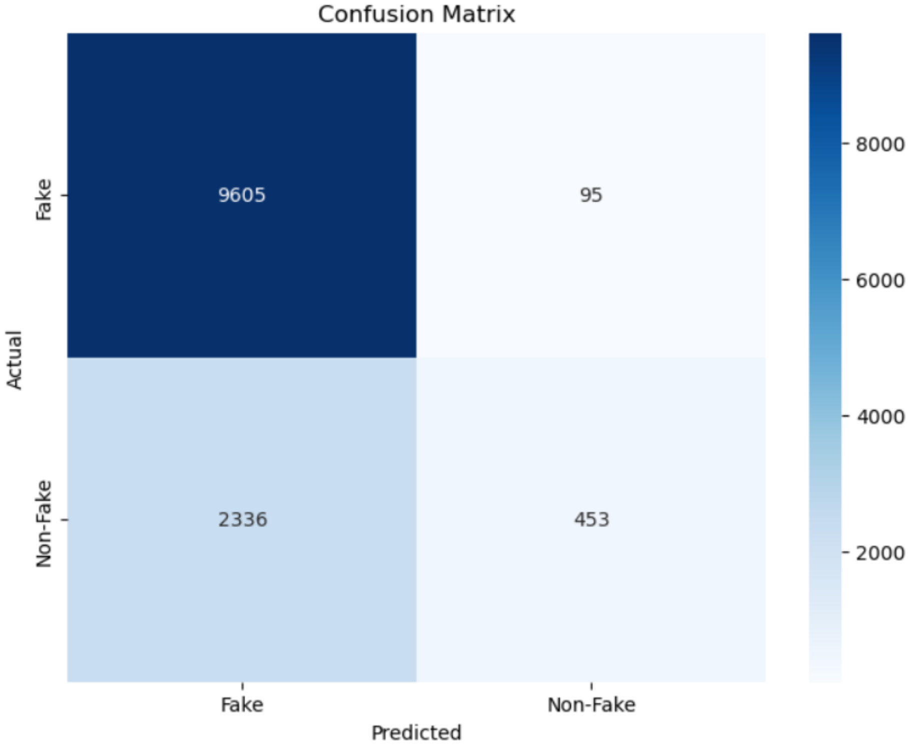


Figure 4. 3 - Confusion Matrix Before EFEFI -SVM

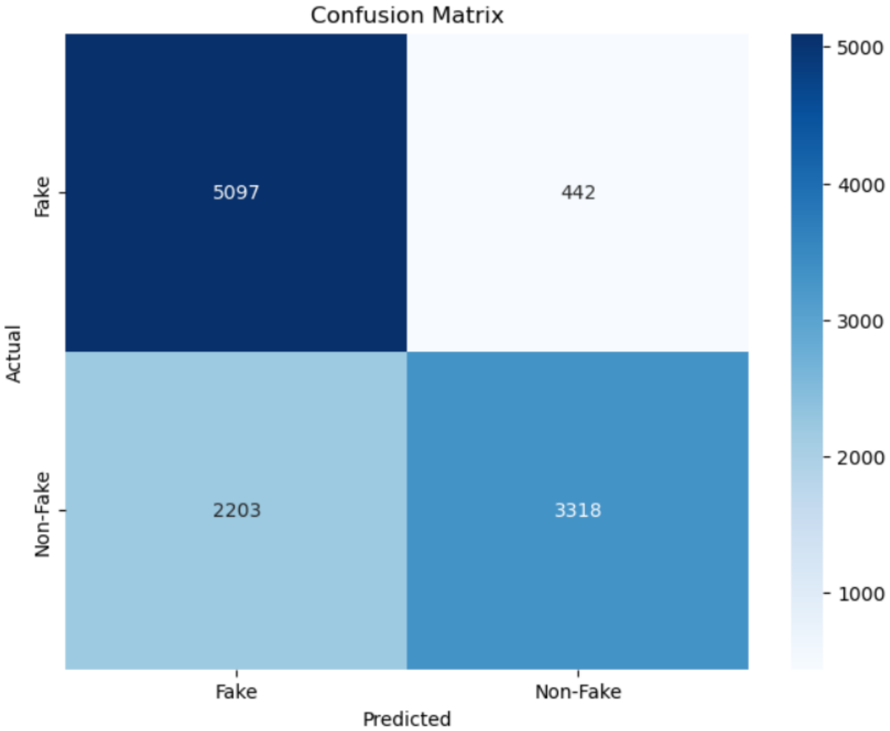
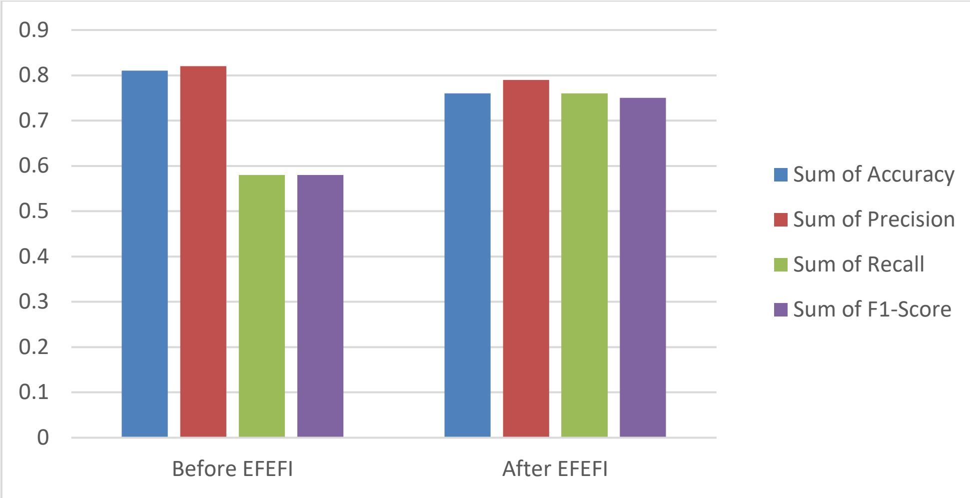


Figure 4. 4 - Confusion Matrix After EFEFI -SVM

After analyzing the confusion matrix, we calculated the accuracy, precision, F1 score, and recall values before and after applying EFEFI. The results are summarized in Table 4.2, which provides a comparison of the performance metrics for the SVM Algorithm. Additionally, Graph 4.2 graphically represents the changes in these metrics before and after applying EFEFI, allowing for a visual understanding of the improvements achieved.

SVM Algorithm	Accuracy	Precision	Recall	F1-Score
Before EFEFI	81 %	82 %	58 %	58 %
After EFEFI	76 %	79 %	76 %	75 %

Table 4. 2 – SVM Comparison Before and After EFEFI



Graph 6. 2 – SVM Comparison Before and After EFEFI

6.1.3 Decision Tree

Below is the confusion matrix using Decision tree before and after using EFEFI depicted in Figure 4.5 and 4.6 respectively.

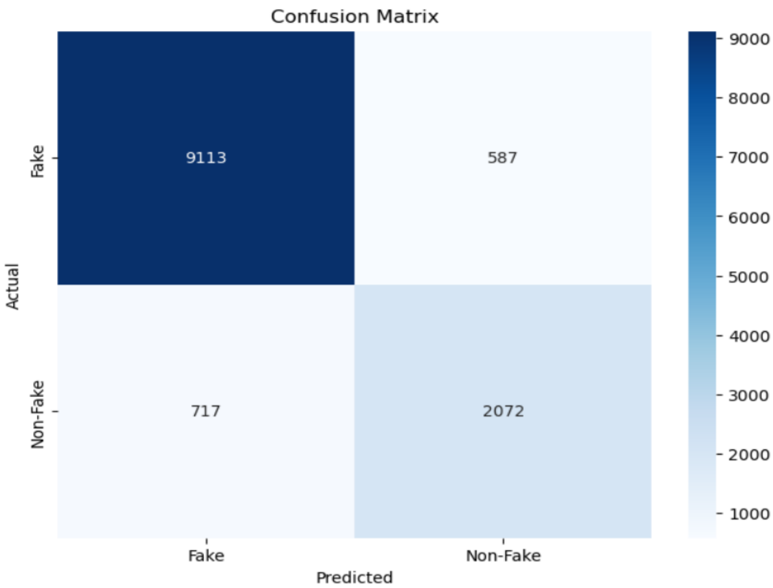


Figure 4. 5 - Confusion Matrix Before EFEFI – Decision Tree

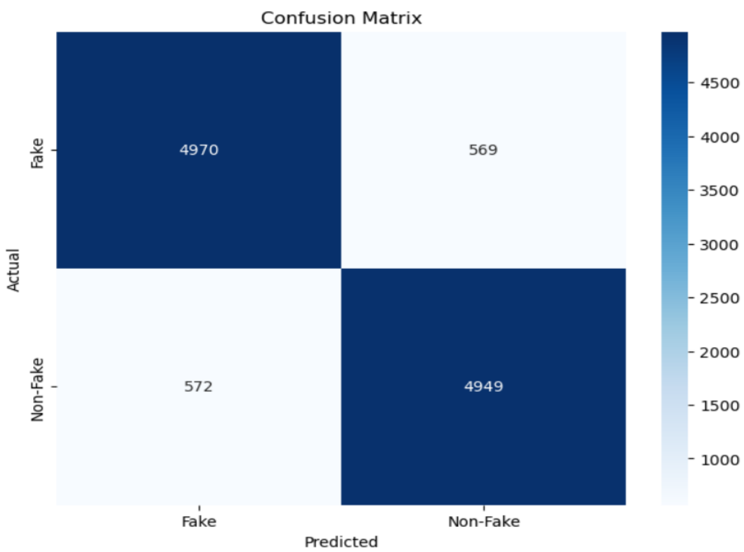
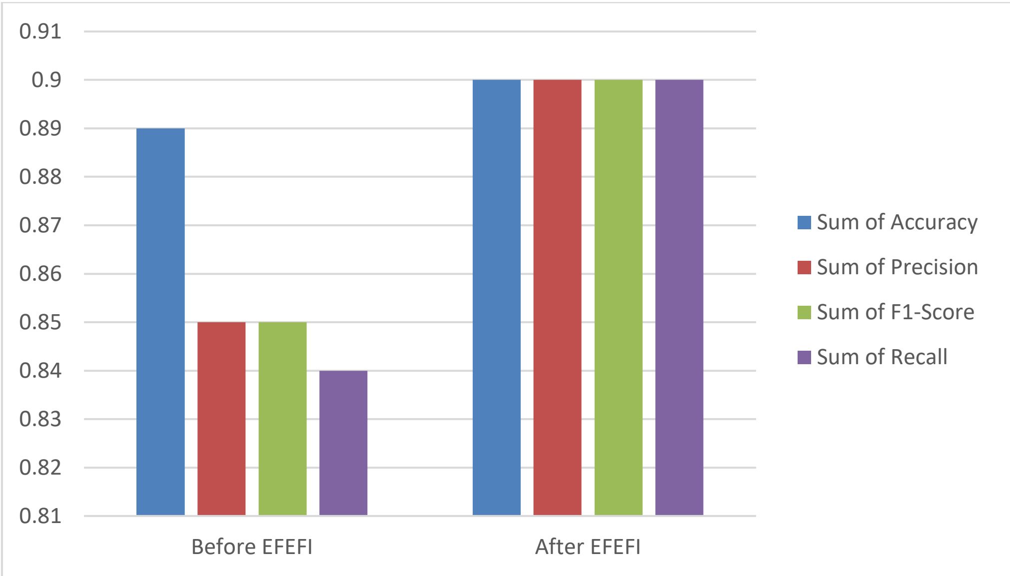


Figure 4. 6 - Confusion Matrix After EFEFI – Decision Tree

Table 4.3 presents the values of accuracy, precision, recall, and F1-score before and after applying EFEFI. The table provides a comprehensive comparison of the performance metrics, showcasing the impact of the EFEFI model. Additionally, Graph 4.3 visually represents the changes in these metrics, allowing for a clearer understanding of the improvements achieved by incorporating EFEFI into the analysis.

Decision Tree Algorithm	Accuracy	Precision	Recall	F1-Score
Before EFEFI	89 %	85 %	84 %	85 %
After EFEFI	90 %	90 %	90 %	90 %

Table 4. 3 – DT Comparison Before and After EFEFI



Graph 4. 3 – DT Comparison Before and After EFEFI

4.1.4 Random Forest

Confusion Matrix using Random Forest algorithm before and after using EFEFI is depicted in figure 4.7 and 4.8 below:

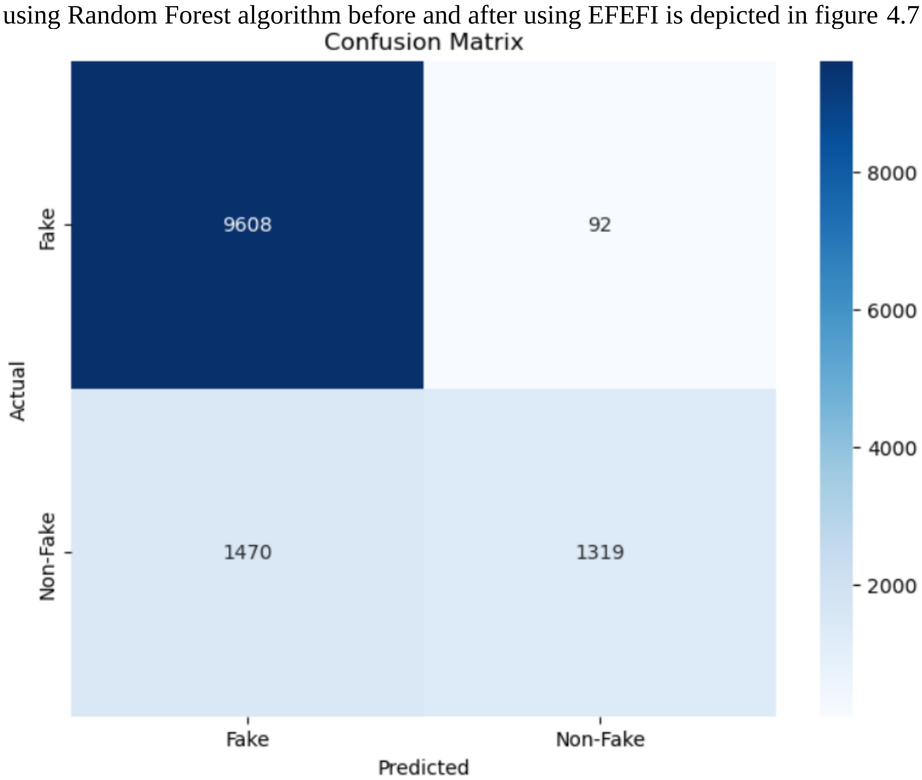


Figure 4. 7 - Confusion Matrix Before EFEFI – Random Forest

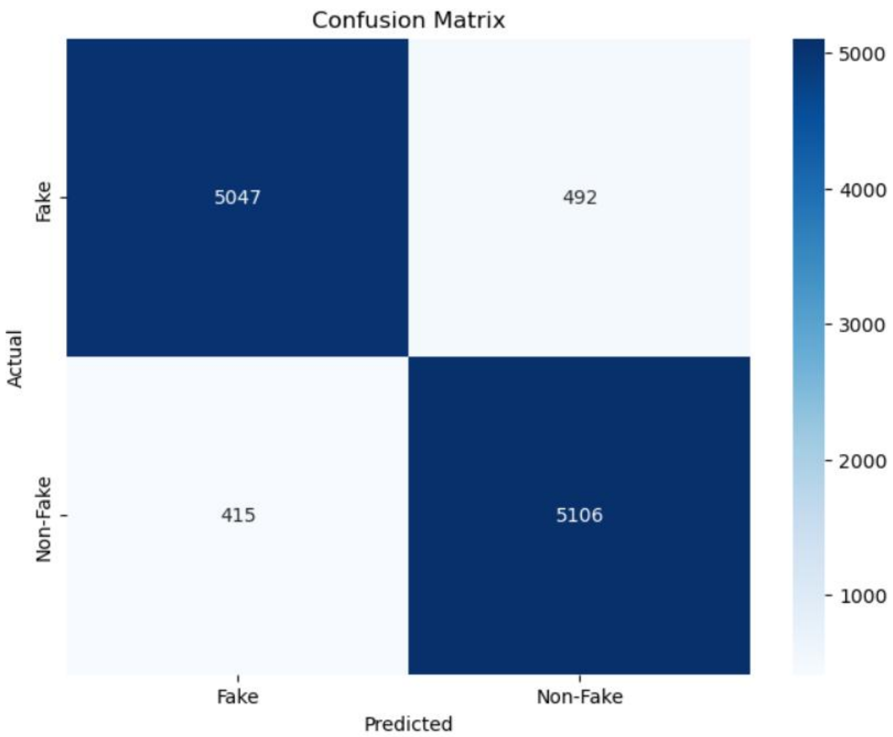
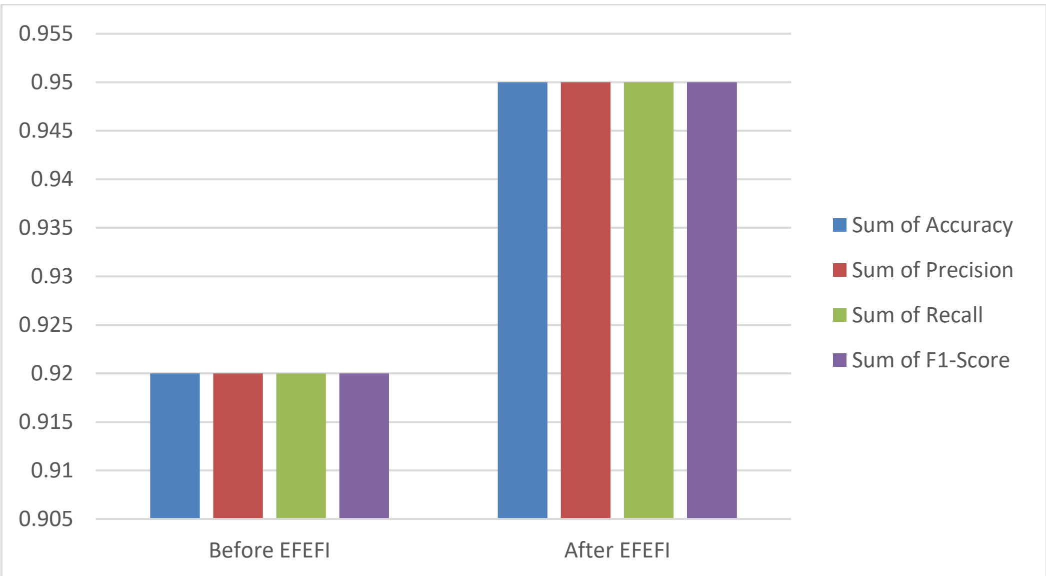


Figure 4. 8 - Confusion Matrix After EFEFI – Random Forest

Based on the confusion matrix above we find accuracy, Precision, F1 score and recall values before and after EFEFI and the same is shown in the Table 4.4 and depicted in graphical form in Graph 4.4.

Random Forest Algorithm	Accuracy	Precision	Recall	F1-Score
Before EFEFI	90 %	90 %	90 %	90 %
After EFEFI	95 %	95 %	95 %	95 %

Table 4. 4 – RF Comparison Before and After EFEFI



Graph 4. 4 – RF Comparison Before and After EFEFI

4.1.5 Voting Ensemble Method

The Confusion Matrix for the Support Vector Machine, both before and after applying EFEFI, is illustrated in Figure 4.9 and Figure 4.10

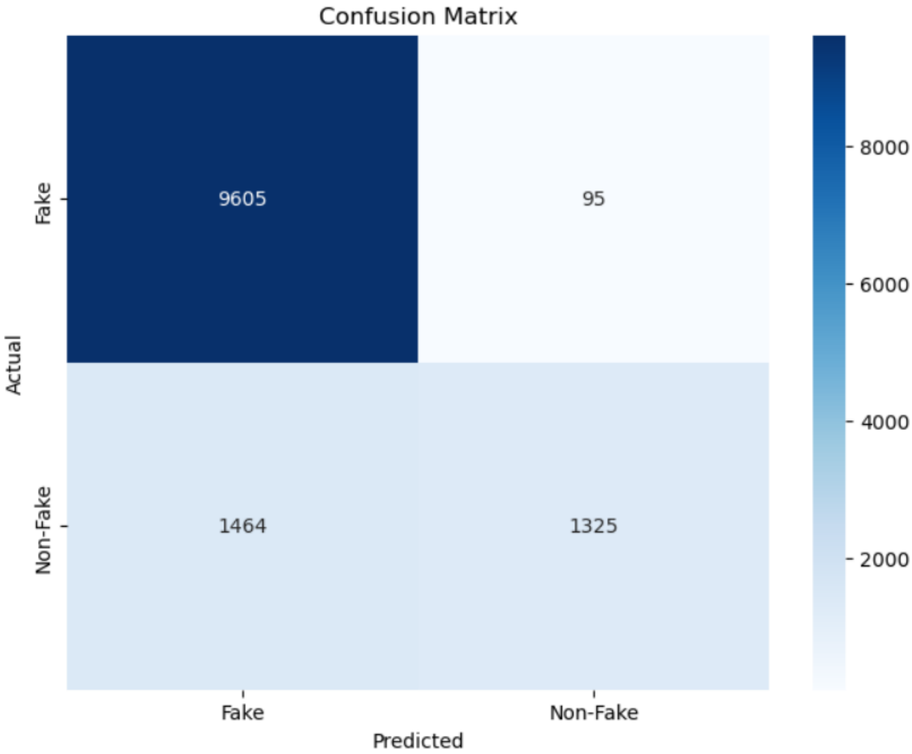


Figure 4. 9 - Confusion Matrix Before EFEFI – EN

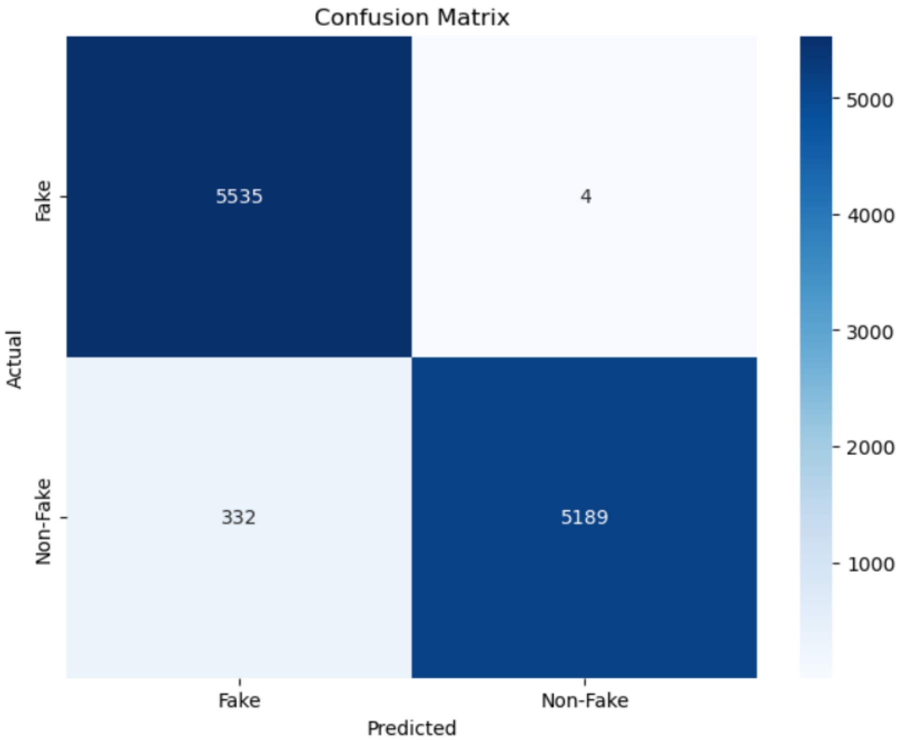
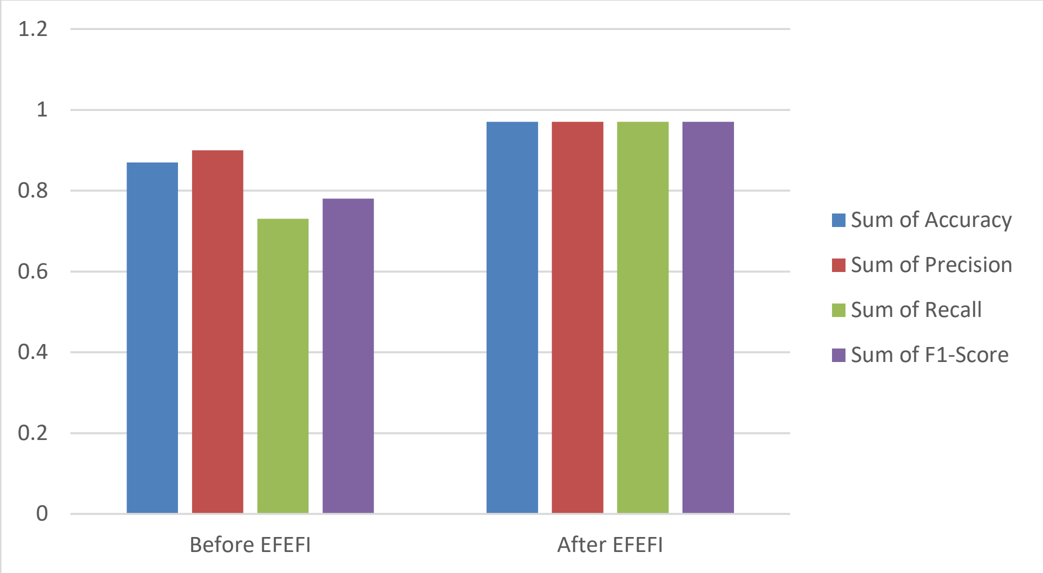


Figure 4. 10 - Confusion Matrix After EFEFI – EN

The accuracy, precision, recall and f-score values of the above confusion matrix can be depicted in tabular form as in Table 4.5 and same can be represented in graphical form as in Graph 4.5.

Ensembe Voting Method	Accuracy	Precision	Recall	F1-Score
Before EFEFI	87 %	90 %	73 %	78 %
After EFEFI	97 %	97 %	97 %	97 %

Table 4. 5 – EN Comparison Before and After EFEFI



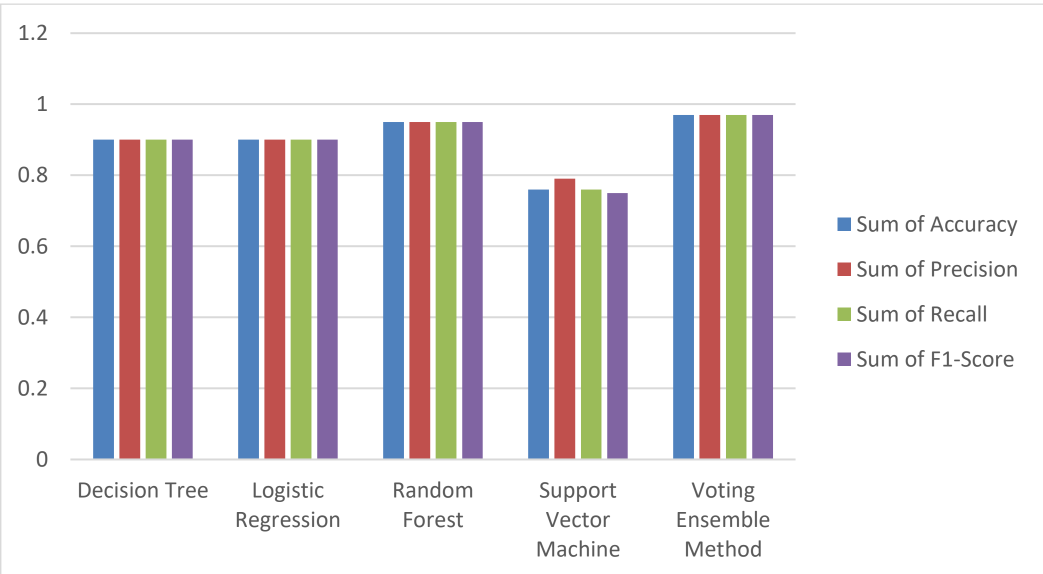
Graph 4. 5 - EN Comparison Before and After EFEFI

4.2 - Comparison of the Algorithms After EFEFI

From the above results it is evident that after EFEFI the efficiency of model improved for all the algorithms. So, in this subsection, our objective is to evaluate and compare the performance of all the algorithms employed in our study after EFEFI in order to determine the best-performing one. By carefully analyzing the results obtained from each algorithm, we aim to identify the model that exhibits the highest accuracy, precision, recall, and F1-score. This thorough comparison will enable us to make an informed decision and select the most effective algorithm for our specific task. The comparison of the different models is presented in Table 4.6, showcasing their respective performance metrics such as accuracy, precision, recall, and F1-score. Additionally, Graph 4.6 visually represents the comparative analysis, providing a graphical representation of the performance of each model. These comprehensive visualizations and tabular data allow for a clear understanding and comparison of the models, aiding in the selection of the best model.

Algorithm	Accuracy	Precision	Recall	F1-Score
Logistic Regression	90%	90%	90%	90%
Support Vector Machine	76%	79%	76%	75%
Decision Tree	90%	90%	90%	90%
Random Forest	95%	95%	95%	95%
Voting Ensemble Method	97%	97%	97%	97%

Table 4. 6 - Comparison Among Models After EFEFI



Graph 4. 6 – Comparison Among Models After EFEFI

Based on the results presented above, it is evident that the Voting ensemble method demonstrates the best performance among the various algorithms evaluated. It exhibits higher accuracy, precision, recall, and F1-score compared to the other models. These findings highlight the effectiveness of the Voting ensemble approach in achieving accurate and reliable predictions. Therefore, based on these results, we can conclude that the Voting ensemble method is the most suitable choice for our task.

5 - Conclusion

In this research, we extensively examined the role of machine learning in the digital realm, with a particular focus on detecting and combating fake news prevalent on social media platforms. Our study delved into various methodologies to effectively identify and address the dissemination of false information. To enhance the accuracy of machine learning algorithms in fake news detection, we put forth an innovative approach called EFEFI (Enhanced Feature Engineering for Improved Accuracy), which integrates hybrid feature engineering techniques. Our comprehensive methodology for fake news detection encompasses several crucial stages, including data collection from Twitter using the platform's API, thorough data preprocessing techniques such as tokenization and lemmatization, and meticulous feature engineering. The hybrid feature engineering approach involves combining diverse features such as TF/IDF (Term Frequency-Inverse Document Frequency), Bag of Words, tweet length, and sentiment analysis. By leveraging these enriched features, we aimed to improve the overall performance of machine learning classifiers. The experimental results exhibited notable enhancements in accuracy after implementing EFEFI.

Among the various machine learning classifiers tested, the ensemble method known as Voting demonstrated superior performance compared to others (97%). This outcome underscores the efficacy of our proposed model in effectively identifying and countering fake news. To ensure the reliability and robustness of our approach, we conducted K-Fold cross-validation, a widely accepted validation technique. This process involved dividing the dataset into k subsets, training the model on a majority of the data while evaluating its performance on the remaining subset. The successful validation of our proposed model further affirms its effectiveness in combating the proliferation of fake news.

In conclusion, our research presents an integrated framework that leverages machine learning algorithms, hybrid feature engineering, and rigorous validation methods to detect and address the circulation of fake news on social media platforms.

References -

[1] T. Zarger and S. Lal, “Machine learning perspective for analysis of geospatial data,” *Journal of Data Acquisition and Processing*, vol. 38, no. 1, 2023.

[2] V. L. Rubin, Y. Chen, and N. K. Conroy, “Deception detection for news: three types of fakes,” *Proceedings of the Association for Information Science and Technology*, vol. 52, no. 1, pp. 1–4, 2015.

[3] M. Siering, J.-A. Koch, and A. V Deokar, “Detecting fraudulent behavior on crowdfunding platforms: The role of linguistic and content-based cues in static and dynamic contexts,” *Journal of Management Information Systems*, vol. 33, no. 2, pp. 421–455, 2016.

[4] C. Cai, L. Li, and D. Zeng, “Detecting social bots by jointly modeling deep behavior and content information,” in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, 2017, pp. 1995–1998.

[5] J. Golbeck *et al.*, “Fake news vs satire: A dataset and analysis,” in *Proceedings of the 10th ACM conference on web science*, 2018, pp. 17–21.

[6] Y. Liu and Y.-F. Wu, “Early detection of fake news on social media through propagation path classification with recurrent and convolutional networks,” in *Proceedings of the AAAI conference on artificial intelligence*, 2018.

[7] J. Ma, W. Gao, and K.-F. Wong, “Rumor detection on twitter with tree-structured recursive neural networks,” Association for Computational Linguistics, 2018.

[8] J. Zhang, B. Dong, and S. Y. Philip, “Fakedetector: Effective fake news detection with deep diffusive neural network,” in *2020 IEEE 36th international conference on data engineering (ICDE)*, IEEE, 2020, pp. 1826–1829.

[9] X. Zhou and R. Zafarani, “Network-based fake news detection: A pattern-driven approach,” *ACM SIGKDD explorations newsletter*, vol. 21, no. 2, pp. 48–60, 2019.

[10] S. Castelo *et al.*, “A topic-agnostic approach for identifying fake news pages,” in *Companion proceedings of the 2019 World Wide Web conference*, 2019, pp. 975–980.

[11] X. Zhou, A. Jain, V. V Phoha, and R. Zafarani, “Fake news early detection: A theory-driven model,” *Digital Threats: Research and Practice*, vol. 1, no. 2, pp. 1–25, 2020.

[12] T. Bian *et al.*, “Rumor detection on social media with bi-directional graph convolutional networks,” in *Proceedings of the AAAI conference on artificial intelligence*, 2020, pp. 549–556.

[13] T. Bian *et al.*, “Rumor detection on social media with bi-directional graph convolutional networks,” in *Proceedings of the AAAI conference on artificial intelligence*, 2020, pp. 549–556.

[14] V. Sabeeh, M. Zohdy, and R. Al Bashaireh, “Enhancing the fake news detection by applying effective feature selection based on semantic sources,” in *2019 international conference on computational science and computational intelligence (CSCI)*, IEEE, 2019, pp. 1365–1370.

[15] G. Guibon, L. Ermakova, H. Seffih, A. Firsov, and G. Le Noé-Bienvenu, “Multilingual fake news detection with satire,” in *International Conference on Computational Linguistics and Intelligent Text Processing*, Springer, 2019, pp. 392–402.

[16] Y.-F. Huang and P.-H. Chen, “Fake news detection using an ensemble learning model based on self-adaptive harmony search algorithms,” *Expert Syst Appl*, vol. 159, p. 113584, 2020.

[17] T. Thaher, M. Saheb, H. Turabieh, and H. Chantar, "Intelligent detection of false information in arabic tweets utilizing hybrid harris hawks based feature selection and machine learning models," *Symmetry (Basel)*, vol. 13, no. 4, p. 556, 2021.

[18] B. Al-Ahmad, A. Al-Zoubi, R. Abu Khurma, and I. Aljarah, "An evolutionary fake news detection method for covid-19 pandemic information," *Symmetry (Basel)*, vol. 13, no. 6, p. 1091, 2021.

[19] F. A. Ozbay and B. Alatas, "Adaptive Salp swarm optimization algorithms with inertia weights for novel fake news detection model in online social media," *Multimed Tools Appl*, vol. 80, no. 26, pp. 34333–34357, 2021.

[20] B. Probierz, J. Kozak, P. Stefański, and P. Juszczuk, "Adaptive goal function of ant colony optimization in fake news detection," in *Computational Collective Intelligence: 13th International Conference, ICCCI 2021, Rhodes, Greece, September 29–October 1, 2021, Proceedings 13*, Springer, 2021, pp. 387–400.

[21] S. Sheikhi, "An effective fake news detection method using WOA-xgbTree algorithm and content-based features," *Appl Soft Comput*, vol. 109, p. 107559, 2021.

[22] A. Sawan and T. Thaher, "Sentiment analysis model for fake news identification in Arabic tweets," in *2021 IEEE 15th international conference on application of information and communication technologies (AICT)*, IEEE, 2021, pp. 1–6.

[23] M. Zivkovic, C. Stoean, A. Petrovic, N. Bacanin, I. Strumberger, and T. Zivkovic, "A novel method for covid-19 pandemic information fake news detection based on the arithmetic optimization algorithm," in *2021 23rd international symposium on symbolic and numeric algorithms for scientific computing (SYNASC)*, IEEE, 2021, pp. 259–266.

[24] M. Khashei and M. Bijari, "A novel hybridization of artificial neural networks and ARIMA models for time series forecasting," *Appl Soft Comput*, vol. 11, no. 2, pp. 2664–2675, 2011.

[25] M. J. C. Samonte, "Polarity analysis of editorial articles towards fake news detection," in *Proceedings of the 2018 1st international conference on internet and e-business*, 2018, pp. 108–112.

[26] A. Jain, A. Shakya, H. Khatter, and A. K. Gupta, "A smart system for fake news detection using machine learning," in *2019 International conference on issues and challenges in intelligent computing techniques (ICICT)*, IEEE, 2019, pp. 1–4.

[27] H. S. Al-Ash, M. F. Putri, P. Mursanto, and A. Bustamam, "Ensemble learning approach on indonesian fake news classification," in *2019 3rd international conference on informatics and computational sciences (ICICoS)*, IEEE, 2019, pp. 1–6.

[28] P. Bharadwaj and Z. Shao, "Fake news detection with semantic features and text mining," 2019.

[29] A. Y. A. Amer and T. Siddiqui, "Detection of covid-19 fake news text data using random forest and decision tree classifiers," *International Journal of Computer Science and Information Security (IJCSIS)*, vol. 18, no. 12, pp. 88–100, 2020.

[30] M. G. Hussain, M. R. Hasan, M. Rahman, J. Protim, and S. Al Hasan, "Detection of bangla fake news using mnb and svm classifier," in *2020 International conference on computing, electronics & communications engineering (iCCECE)*, IEEE, 2020, pp. 81–85.

[31] D. Varshney and D. K. Vishwakarma, "A review on rumour prediction and veracity assessment in online social network," *Expert Syst Appl*, vol. 168, p. 114208, 2021.

[32] D. Varshney and D. K. Vishwakarma, "A review on rumour prediction and veracity assessment in online social network," *Expert Syst Appl*, vol. 168, p. 114208, 2021.

[33] T. Davidson, D. Warmsley, M. Macy, and I. Weber, "Automated hate speech detection and the problem of offensive language," in *Proceedings of the international AAAI conference on web and social media*, 2017, pp. 512–515.