GLOBAL JOURNAL
OF COMPUTER SCIENCE AND TECHNOLOGY - D: NEURAL & AI
Volume 26 | 2026 Article ID: GJCST-259604 Language: English

Enhancing Feature Selection for Fake News Detection using A Hybrid Genetic Algorithm and Particle Swarm Optimization Approach

Online First

Nikita Garg [§ θ *]
[θ] Research Scholar
ORCID 0009-0009-1418-2611
[*] Corresponding Author

Dr. Pritam Singh Negib [§ ξ]
[ξ] Assistant Professor
ORCID 0009-0004-5703-6212

Nischay Garg [‡ θ]
[θ] M.Tech Student

[§] Department of Computer Science & Engineering, HNB Garhwal University, Srinagar, India
[‡] Department of Computer Science and Engineering, Dr. APJ Abdul Kalam University, Lucknow, India

RECEIVED	ACCEPTED	ONLINE PUBLISHED	PEER REVIEW
2026-08-01	2026-08-11	2026-08-21	Double Blind

Abstract

Fake news has emerged as a major concern in today's digital era, spreading rapidly and influencing various aspects of society, including mental well-being and public opinion. This study focuses on addressing this issue by proposing an effective framework for detecting fake news. The research utilizes the FakeNewsNet dataset, beginning with thorough data pre-processing to ensure quality and consistency. Features were then extracted using the TF-IDF (Term Frequency-Inverse Document Frequency) technique, which is widely used for textual data analysis. To further enhance the dataset, a hybrid approach combining Genetic Algorithm and Particle Swarm Optimization was employed for feature selection. This hybrid method ensures that only the most relevant and impactful features are retained, optimizing the overall performance of the detection system. The selected features were used to train different machine-learning models. Among these, Logistic Regression delivered 99% accuracy, a significant result. However, three other machine learning algorithms outperformed it, achieving even higher accuracy, highlighting their superior ability to classify fake news accurately. This research aims to contribute to the ongoing fight against fake news by providing a reliable and efficient detection system that can help mitigate its negative effects on society.

Feature Selection | Hybrid Approach | Genetic Algorithm | Particle Swarm Optimization | Fake News

AI USE STATEMENT
No generative AI was used for analysis or results.

FUNDING
The present manuscript has no funding source to declare.

CONFLICT OF INTEREST
The authors declare no conflict of interest.

DATA AVAILABILITY
Not applicable for this article.

ETHICS
No ethics committee approval was required for this article type.

CONSENT
Not applicable.

TRIAL REG.
Not applicable.

Crossref DOI: 10.34257/GJCSTD259604

How to Cite: Garg et al. (2026). Enhancing Feature Selection for Fake News Detection using A Hybrid Genetic Algorithm and Particle Swarm Optimization Approach. Global Journal of Computer Science and Technology. Ahead of Print. DOI: 10.34257/GJCSTD259604

LICENSE
© 2026 Global Journals. Open-access article under CC BY-NC-ND 4.0 International License.

AR Experience Web Page DOI
Print ISSN 0975-4350
Online ISSN 0975-4172
Under the strict compliance and defined process of

METADATA CONTINUATION

AUTHOR CONTACT QR LEDGER

Nikita Garg[§θ*]

Dr. Pritam Singh Negib[§ξ]

Nischay Garg[‡θ]

ARCHIVAL RECORD

GJCST · Vol 26 · 2026

Article ID GJCST-259604 · DOI 10.34257/GJCSTD259604

Print ISSN 0975-4350 · Online ISSN 0975-4172

Global Journal of Computer Science and Technology - Section D
GJCST

Enhancing Feature Selection for Fake News Detection using A Hybrid Genetic Algorithm and Particle Swarm Optimization Approach

Nikita Garg[§ ϑ * iD], Dr. Pritam Singh Negib[§ ξ iD], and Nischay Garg[‡ θ]

Affiliations
§ Department of Computer Science & Engineering, HNB Garhwal University, Srinagar, India
‡ Department of Computer Science and Engineering, Dr. APJ Abdul Kalam University, Lucknow, India

Qualifications / Designations
ϑ Research Scholar
ξ Assistant Professor
θ M.Tech Student

Abstract

Fake news has emerged as a major concern in today's digital era, spreading rapidly and influencing various aspects of society, including mental well-being and public opinion. This study focuses on addressing this issue by proposing an effective framework for detecting fake news. The research utilizes the FakeNewsNet dataset, beginning with thorough data pre-processing to ensure quality and consistency. Features were then extracted using the TF-IDF (Term Frequency-Inverse Document Frequency) technique, which is widely used for textual data analysis. To further enhance the dataset, a hybrid approach combining Genetic Algorithm and Particle Swarm Optimization was employed for feature selection. This hybrid method ensures that only the most relevant and impactful features are retained, optimizing the overall performance of the detection system. The selected features were used to train different machine-learning models. Among these, Logistic Regression delivered 99% accuracy, a significant result. However, three other machine learning algorithms outperformed it, achieving even higher accuracy, highlighting their superior ability to classify fake news accurately. This research aims to contribute to the ongoing fight against fake news by providing a reliable and efficient detection system that can help mitigate its negative effects on society.

Keywords: *Feature Selection, Hybrid Approach, Genetic Algorithm, Particle Swarm Optimization, Fake News*

*** Corresponding Author**
Nikita Garg

DOI
10.34257/GJCSTD259604

1. Introduction

The advent of the digital age has revolutionized the way news is produced, accessed, and consumed. Social media platforms, blogs, and other online channels have become primary sources of information, offering unparalleled accessibility. However, this transformation has introduced a significant challenge—the rapid proliferation of fake news. Fake news, defined as deliberately fabricated or misleading information intended to deceive readers, poses severe societal risks. Its potential to manipulate public opinion, distort political processes, and deepen social divides is well-documented, particularly during critical events like elections, public health crises, and social movements [2]. The urgency to detect and mitigate the spread of fake news necessitates the development of efficient and accurate detection frameworks. Detecting fake news involves analyzing large-scale textual data to differentiate between authentic and fabricated content. A crucial step in this process is feature selection, which involves identifying the most relevant and impactful features from the dataset [3]. Effective feature selection not only improves the accuracy of machine learning models but also reduces computational overhead. However, in high-dimensional textual datasets, traditional feature selection methods often struggle to identify meaningful features due to their inherent limitations, including inefficiency and susceptibility to suboptimal performance in dynamic environments [4]. In this study, advanced optimization techniques are employed to address the challenges of feature selection in fake news detection. These techniques fall under the umbrella of Evolutionary Computation, which encompasses a range of optimization techniques inspired by natural processes, such as biological evolution and social behavior. These techniques are generally divided into two main categories: Evolutionary Algorithms and Swarm Intelligence. Evolutionary Algorithms are a type of optimization technique based on principles of natural evolution [5]. Swarm Intelligence, on the other hand, refers to optimization techniques inspired by collective behavior in natural systems, such as the behavior of social animals [6]. Both of these categories are designed to solve complex optimization problems by mimicking natural processes, though they operate in different ways. Evolutionary Algorithms, inspired by natural evolutionary processes, have emerged as a powerful tool for feature selection, offering adaptive and robust solutions for complex problems [7]. These algorithms mimic biological processes such as selection, mutation, and crossover to evolve a population of potential solutions over time. Genetic Algorithms, a well-known subclass of EAs, have been particularly successful in feature selection tasks due to their ability to explore large and complex search spaces efficiently [8]. In Genetic Algorithm, a population of candidate solutions is evolved over successive generations, with the fittest individuals selected for reproduction based on their ability

to meet the problem's objectives. This iterative process of selection, crossover, and mutation enables Genetic Algorithm to explore diverse solutions and avoid local optima, making them well-suited for feature selection in high-dimensional datasets [9]. Despite their strengths, Genetic Algorithms face certain limitations [10]. One significant challenge is their tendency to prematurely converge to suboptimal solutions, especially when the search space is large or the problem is highly complex [10]. This issue arises because Genetic Algorithm may focus on a limited subset of features early in the search process, reducing the diversity of the population and hindering further exploration of the solution space.

To address the limitations of Genetic Algorithm, Swarm Intelligence techniques, particularly Particle Swarm Optimization (PSO), have gained popularity for feature selection. Inspired by the collective behavior of social organisms such as birds or fish, Particle Swarm Optimization simulates the movement of particles in a search space, with each particle representing a potential solution. Particles adjust their positions based on both their own best-known solution and the best-known solution in the swarm [11]. Particle Swarm Optimization has been successfully applied to a wide range of optimization problems, including feature selection, due to its simplicity, ease of implementation, and ability to converge rapidly to high-quality solutions. However, while Particle Swarm Optimization excels in fine-tuning solutions and exploring local neighborhoods, it can become trapped in local optima, especially in high-dimensional and complex problems [12]. This issue can lead to suboptimal feature selection, where the algorithm fails to identify the most relevant features that truly contribute to the performance of the classification model. Recognizing the complementary strengths of Genetic Algorithm and Particle Swarm Optimization, hybrid approaches combining both techniques have been proposed to overcome the individual limitations of each. In a hybrid Genetic Algorithm-Particle Swarm Optimization framework, the Genetic Algorithm is used to explore a broad solution space, while Particle Swarm Optimization fine-tunes the selected solutions for optimal feature sets [12]. This combination leverages the global search capability of Genetic Algorithm and the local search efficiency of Particle Swarm Optimization, leading to more robust feature selection and improved model performance. Hybrid Genetic Algorithm-Particle Swarm Optimization frameworks have been shown to significantly enhance the accuracy of classification models by identifying the most relevant features while reducing dimensionality and computational cost [13].

This research builds on these advancements by proposing a hybrid Genetic Algorithm-Particle Swarm Optimization framework specifically designed for fake news detection, a task characterized by high-dimensional, noisy textual data. By integrating both Evolution Algorithm and Swarm Intelligence, the proposed framework aims to optimize feature selection for fake news detection, improving both the efficiency and accuracy of machine learning models. The dataset used for this research is the FakeNewsNet dataset [1], which is publicly available on Kaggle and widely used within the research community. This dataset contains real and fake news articles and is part of several popular fake news detection datasets such as LIAR, PHEME, BUZZFEED, and CIFAR, all of which are available on Kaggle. After collecting the data, pre-processing steps were performed to structure the dataset appropriately, ensuring it was ready for further analysis. The dataset includes the following columns: title, text, subject, date, and label. In this research, we focused on the title, text, subject, and label columns, where the label column represents the target variable (fake or real news). Once the data was prepared, feature extraction was performed using the TF-IDF (Term Frequency-Inverse Document Frequency) method, which is particularly effective for text data. After extracting the features, 5000 features were initially selected. A hybrid Genetic Algorithm-Particle Swarm Optimization approach was then used to optimize the selection, resulting in 3010 selected features. These selected features were used to train four different machine learning algorithms, and the performance of each model was evaluated. The performance analysis revealed that Logistic Regression, while achieving an accuracy of 99%, had a much lower precision and recall compared to the other algorithms. This suggests that while Logistic Regression may achieve high accuracy, it is not as effective in identifying fake news articles compared to other algorithms. The hybrid Genetic Algorithm-Particle Swarm Optimization approach demonstrated that evolutionary algorithms are an efficient and powerful tool for feature selection in fake news detection. Compared to traditional methods, which tend to be less effective in selecting the right features, the hybrid approach provided a more accurate and robust solution, significantly enhancing the accuracy of fake news detection models.

This research contributes to the field by introducing the following innovations: Development of a Hybrid Genetic Algorithm-Particle Swarm Optimization Framework: This framework leverages the complementary strengths of Genetic Algorithms and Particle Swarm Optimization to improve feature selection in high-dimensional datasets. Application to Fake News Detection: The hybrid approach was applied to the FakeNewsNet dataset, a real-world problem that benefits from advanced optimization techniques for more accurate and efficient detection. Comprehensive Evaluation: The framework was evaluated using machine learning models, showing its effectiveness in improving fake news detection accuracy and proving its utility in high-dimensional data problems. This research addresses the critical issue of fake news detection and highlights the broader potential of hybrid evolutionary algorithms for solving complex, high-dimensional data problems. By enhancing the feature selection process, the hybrid framework provides a scalable and effective solution that can be applied across a variety of domains, improving decision-making processes and reducing the spread of misinformation in society.

The paper is organized into six sections to ensure clarity and coherence. Section 2 focuses on the literature review, providing a comprehensive analysis of previous work related to this study. Section 3 introduces the proposed model, detailing the framework and its application to fake news detection. Section 4 is dedicated to the methodology, explaining the steps involved data pre-processing to model evaluation. Section 4 presents the experimental analysis and results, showcasing the implementation and performance evaluation of the model. Finally, Section 6 concludes the paper with key insights and outlines potential directions for future work, paving the way for further advancements in this field.

2. Literature Review

Fake news detection has become an essential challenge in the digital age, driven by the rapid spread of misinformation across various online platforms. Early detection methods mainly relied on machine learning algorithms that classified news articles based on content-related features like word frequencies, linguistic patterns, and metadata. With the increasing complexity of data, however, there has been a shift toward using more advanced techniques that incorporate both natural language processing and machine learning algorithms to detect fake news. Ahmed et al. highlighted the potential of machine learning models like Random Forests,

Support Vector Machines, and deep learning networks for fake news detection [20]. These techniques can effectively handle large-scale textual datasets, but one challenge that remains is how to identify the most relevant features that contribute to classification. Feature selection plays a pivotal role in fake news detection, as it helps reduce dimensionality, enhance model performance, and improve computational efficiency by selecting only the most discriminative features. The features typically include textual attributes such as n-grams, sentiment scores, linguistic features, and various textual representations like TF-IDF or Word2Vec embeddings. To improve the classification accuracy of fake news models, feature selection methods such as filter, wrapper, and embedded methods have been widely adopted. Filter methods evaluate features based on statistical tests or information gain, while wrapper methods evaluate subsets of features by training a machine learning model and selecting the best subset based on model performance. Embedded methods, on the other hand, select features during the model training process itself, as seen in decision tree-based models like Random Forests. Among the various feature selection techniques, Genetic Algorithms have been widely used due to their global search capabilities and ability to explore large search spaces efficiently. Pudil et al. [14] were among the first to apply GA for feature selection, and since then, this approach has gained popularity in various fields, including fake news detection. In particular, Zhou et al. [16] showed how GA could identify relevant features for fake news detection, highlighting its ability to reduce overfitting and improve model accuracy. GAs operate by simulating the process of natural selection, where a population of potential feature subsets is evolved over generations to identify the best solution. This makes GA particularly suitable for complex problems where the feature space is large, and redundant or irrelevant features may exist.

In addition to GA, Particle Swarm Optimization has also been widely used for feature selection due to its simplicity and ability to converge quickly toward optimal solutions. Kennedy and Eberhart [11] introduced PSO, and it has since been applied to many optimization problems, including feature selection. Jiang et al. [15] demonstrated the use of PSO for selecting the most relevant features in fake news detection, showing that PSO could reduce the dimensionality of feature sets while maintaining or even improving classification accuracy. The key advantage of PSO lies in its ability to balance exploration and exploitation in the search space, ensuring that the algorithm converges efficiently toward a good solution. To further enhance feature selection performance, hybrid approaches combining GA and PSO have been proposed. These hybrid approaches aim to combine the strengths of both algorithms: GA's global search capabilities and PSO's quick convergence properties. By combining GA and PSO, researchers have been able to develop more efficient feature selection methods that improve the performance of fake news detection models. Deb et al. proposed a hybrid GA-PSO approach for image classification, demonstrating its ability to enhance classification accuracy and reduce computation time [18]. Similarly, Zhou et al. applied a hybrid GA-PSO method to fake news detection, achieving improved feature selection and better overall classification performance [17]. Their results indicated that the hybrid approach outperformed traditional GA or PSO-based methods, as it was better able to explore the search space and fine-tune the selected features.

Once feature selection is completed, various machine learning algorithms can be used for the classification of real and fake news. Algorithms like Random Forests, SVM, and deep learning techniques such as Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) are commonly used for fake news detection. These algorithms can be significantly improved when combined with optimized feature selection techniques like GA and PSO. For example, Ruchansky et al. used deep learning models, including CNNs, to classify fake news, integrating textual features with user behavior data [19]. In addition, Rashid et al. explored the effectiveness of ensemble learning methods for fake news detection and found that combining multiple models improved performance [32]. Integrating feature selection with machine learning algorithms such as these can lead to higher accuracy and more reliable fake news detection systems. Fake news detection has become increasingly challenging with the rise of multimodal data, which combines text, images, and other forms of information. As noted by Li et al., multimodal data fusion offers a more holistic approach to identifying fake news [26]. However, the integration of different modalities leads to a surge in the number of features, which necessitates the use of feature selection techniques. By focusing on the most informative features, feature selection not only improves model efficiency but also reduces redundancy, enhancing overall performance. Numerous studies highlight the importance of feature selection in improving fake news detection. Akinyemi et al. demonstrated that using Genetic Algorithms for feature selection alongside deep learning models significantly boosted model performance by eliminating irrelevant features [40]. Similarly, Kong et al. showed that combining Term Frequency-Inverse Document Frequency (TF-IDF) with Genetic Algorithm-based feature selection resulted in enhanced performance, further emphasizing the importance of refining the feature set [30]. While tree-based models like Random Forest and Gradient Boosting are commonly used in fake news detection due to their robustness, they often lack built-in feature selection components. Kaliyar et al. found that optimized ensemble models were effective at distinguishing real from fake news, but their performance could have been further improved with the addition of feature selection techniques, particularly Genetic Algorithms [27].

Alhariri et al. applied Random Forest for fake news detection in a dataset related to COVID-19 [31]. Although their model performed well, they observed that incorporating feature selection could have enhanced its efficiency by removing redundant or irrelevant features. Similarly, Ahmed et al. explored different machine learning algorithms for fake news detection but did not implement feature selection, which could have further refined the model's performance. For short text datasets, such as tweets, feature selection becomes even more critical. Gupta et al. encountered challenges with large datasets and low accuracy but were able to improve model performance through feature extraction [38]. However, they lacked formal feature selection, which could have helped further reduce dimensionality and improve the effectiveness of fake news detection. Textual content analysis is a common technique for fake news detection, as explored by Wynne et al., who used word and character n-grams combined with machine learning classifiers [28]. Their model showed strong results, but incorporating feature selection methods like Genetic Algorithms could have refined the feature set and potentially improved predictive accuracy. Multimodal approaches that combine text, images, and videos for fake news detection have also been explored. Alshariah et al. used Convolutional Neural Networks (CNNs) and transfer learning with AlexNet to identify fake images [21], while Giachanou et al. demonstrated that combining text and image features using BERT-Base and VGG-16 led to improved detection accuracy [29]. Similarly, Hamid et al. investigated misinformation

related to COVID-19 using Graph Neural Networks, stressing the need for further research on integrating GNNs with Natural Language Processing techniques to enhance fake news detection [39]. While deep learning techniques show promise for fake news detection, the integration of feature selection methods, such as Genetic Algorithms, can significantly improve model performance. Additionally, the inclusion of feature selection in multimodal approaches can enhance both efficiency and accuracy, helping to tackle the growing complexity of data in fake news detection.

3. Proposed Model

Fake news has become a significant issue in contemporary society, and developing effective models for its detection is crucial. In this research, a model for detecting fake news was created using the FakeNewsNet dataset obtained from Kaggle. After gathering the data, it was pre-processed and organized into a structured format to facilitate further analysis. The dataset contains several columns, such as title, text, subject, date, and label. The title, text, and subject columns were combined, which was then processed using TF-IDF (Term Frequency-Inverse Document Frequency) to extract relevant features. Once the features were extracted, a hybrid technique combining Particle Swarm Optimization and Genetic Algorithms was applied to select the most pertinent features. This hybrid method helps identify the most significant features for fake news detection. After selecting the features, the data was classified using four machine learning algorithms. The performance of each algorithm was assessed using standard evaluation metrics, such as accuracy, precision, recall, and F1-score, to determine the effectiveness of the model. The entire process, from data pre-processing to performance evaluation, is depicted in Figure 1, where the various steps are clearly illustrated, offering an easily understandable representation of the approach used for fake news detection.

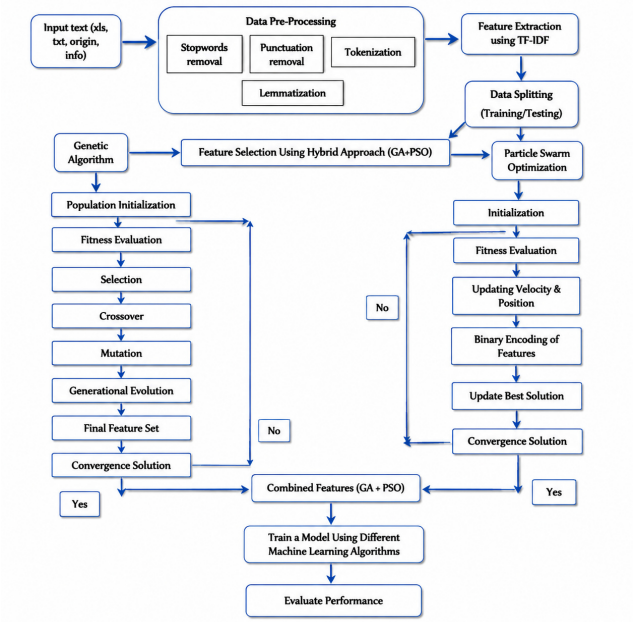

Figure 1. Proposed model of the study and this approach highlights the effectiveness of combining linguistic feature extraction with advanced optimization and classification techniques for fake news detection

4. Methodology

4.1. Data Collection and Visualization

In this study, we used the FakeNewsNet dataset, which is a widely recognized resource available on Kaggle. The dataset is divided into two categories: one for true news and the other for fake news. These categories were combined into a single dataset and labeled as '1' for true news and '0' for fake news to facilitate binary classification. To ensure reliable evaluation, the dataset was split into training and testing sets in an 80:20 ratio, maintaining a balanced distribution of true and fake news across both subsets. This division supports effective model training and performance assessment. The fake news subset contains 23,481 entries across five columns, while the true news subset has 21,417 entries, structured in the same way. After merging the two subsets, the final dataset includes 44,898 records and five columns. For analysis, we focused on four key columns: title, text, subject, and label. To provide a clearer understanding, Figure 2 visualizes a word cloud displaying the most frequent terms in the fake news subset, while Figure 3 illustrates key terms from the true news subset. The dataset consist of the following columns [1]:

- Title: This column contains the headline or title of each news article, summarizing its main theme or content in a concise manner.
- Text: The main body or content of the news article.
- Subject: The category or topic of the news article, such as politics, business, or entertainment.
- Date: The publication date of the news article.
- Label: A binary label representing the authenticity of the news:
  - 1 for true news.
  - 0 for fake news.

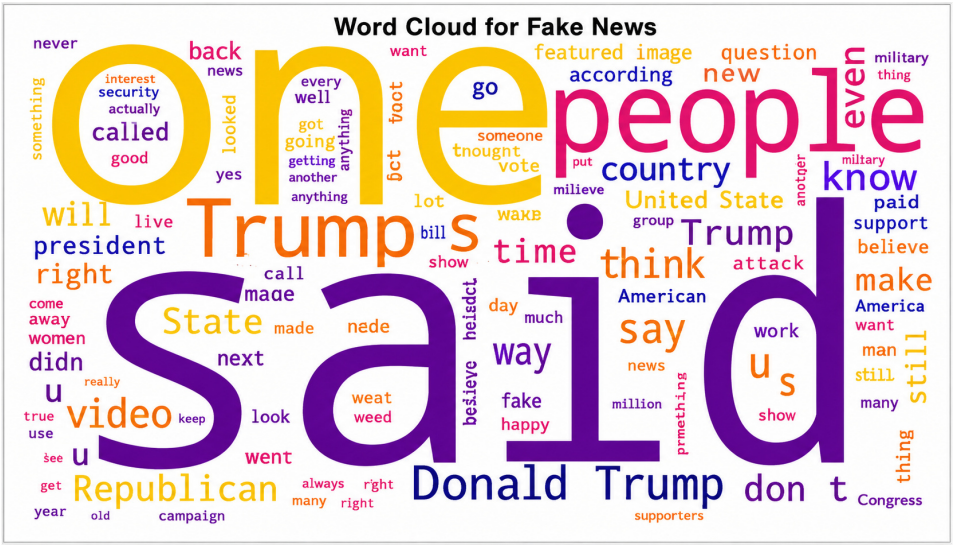

Figure 2. Visualizes a word cloud highlighting the most frequent terms in the fake news subset

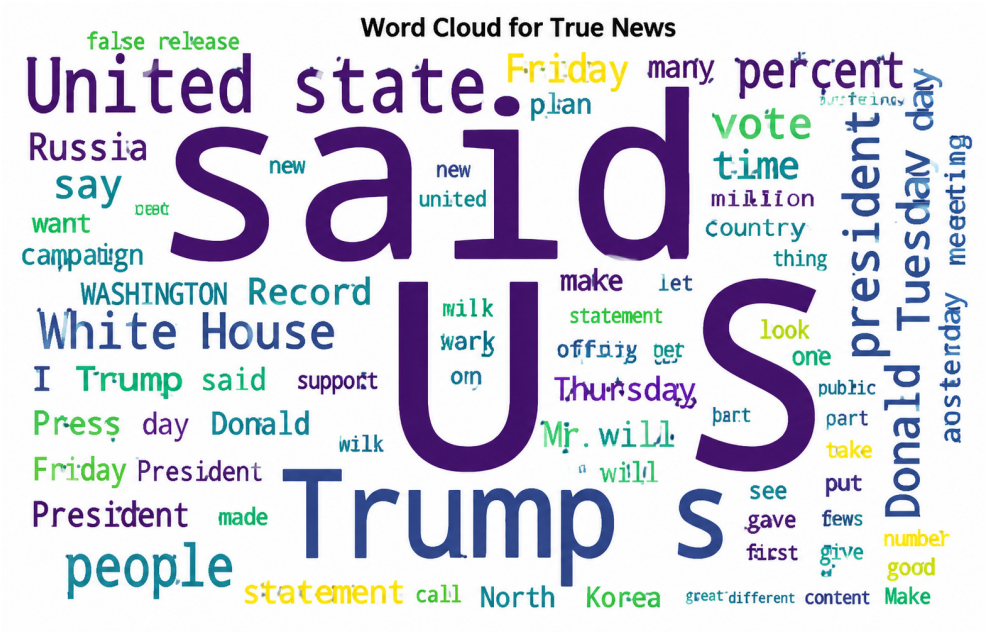

Figure 3. This diagram represents the key terms from the true news subset

4.2. Data Pre-Processing

After collecting the data, it is essential to pre-process it into a structured format to facilitate further processing. In this step, the title, text, and subject columns were merged into a single dataset, and then pre-processing was performed. The output of this pre-processing step is displayed in Figure 4 as a word cloud, highlighting the most frequently occurring terms. During the pre-processing phase, several important tasks were carried out using Python libraries like Pandas, NLTK (Natural Language Toolkit), and re (regular expressions). First, stopwords, URLs, punctuation marks, and other irrelevant content were removed. The NLTK library was used to remove stopwords, while regular expressions were used for handling URLs and punctuation. To ensure uniformity, all the text was converted to lowercase using the `str.lower()` function from Pandas. This step is crucial to avoid treating the same word differently due to variations in capitalization, which can result in redundant data and increased processing time. Reducing such redundancies not only conserves memory but also enhances the accuracy and efficiency of the model. Tokenization, a key step in text pre-processing, was carried out using NLTK's `word_tokenize` function, which splits the text into smaller, manageable units (tokens). These steps were applied to prepare the data for the next stages of analysis and model building, ensuring consistency and efficiency throughout the process. Below are the key steps used in pre-processing the data:

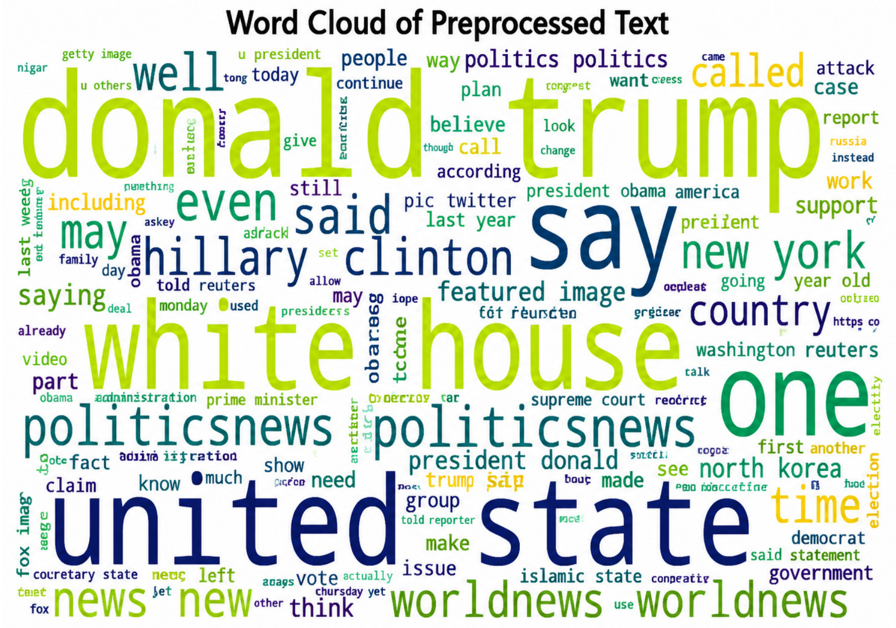

Figure 4. This diagram showcases a word cloud of data pre-processing

4.2.1. Stopwords Removal

Stopwords, such as "the," "and," and "is," are common words that do not provide valuable information for analysis. These words were removed using the NLTK library to eliminate unnecessary noise, allowing the model to focus on the more meaningful content in the text, which helps improve the accuracy of predictions.

4.2.2. Removal of URLs and Punctuation

URLs and punctuation marks do not contribute to understanding the text's content, so they were removed using regular expressions (re). This ensures that the dataset contains only the relevant text, making the processing more efficient and improving the quality of the input data for model training.

4.2.3. Converting Text to Lowercase

To standardize the data, all text was converted to lowercase using Pandas' `str.lower()` method. This is important because it ensures words like "Fake" and "fake" are treated as the same, preventing the model from considering them as separate words, which helps reduce redundancy and unnecessary memory usage.

4.2.4. Tokenization

Tokenization involves splitting the text into smaller, meaningful units (tokens), typically words. This step was carried out using NLTK's `word_tokenize()` function. It breaks the text into individual words or tokens, which can then be used for further analysis, such as feature extraction and classification.

4.3. Feature Extraction using TF-IDF

After completing data pre-processing, features were extracted using TF-IDF vectorization, a statistical method that transforms textual data into a numerical format suitable for machine learning algorithms. TF-IDF assesses how frequently a word appears in a document, helping to evaluate its importance in distinguishing content across a dataset. It is widely used in text mining and natural language processing to extract meaningful features from text data. Selecting 5000 features ensures a more detailed representation of the dataset, capturing a wider range of important patterns and relationships. Using fewer features might risk missing key information while selecting too many could lead to overfitting and increased computational complexity. The choice of 5000 features aims to strike a balance between maintaining important data and ensuring model efficiency, avoiding both underfitting and overfitting. In this study, the title, text, and subject columns were processed using TF-IDF, converting the textual data into a numerical format. The resulting TF-IDF matrix had dimensions of (44898, 5000). Figure 5 illustrates the TF-IDF scores for the top 10 words from the dataset, highlighting the most significant terms based on their frequency and importance in the text and Figure 6 displays a heatmap representing the extracted features derived from the TF-IDF method. This visualization provides an insightful overview of the most significant features in the dataset, highlighting their relative importance and frequency across the documents. The heatmap helps to better understand the distribution and significance of the terms extracted through TF-IDF, offering a clearer picture of the features that contribute most to the analysis. These scores help identify critical features within the data. The transformation reduced the data's dimensionality while preserving essential features for subsequent stages, such as feature selection. The TF-IDF technique highlights terms that help differentiate between documents while reducing the impact of commonly occurring terms. This transformation allows the text to be represented numerically, making it better suited for subsequent analysis and processing.

TF-IDF Formula [33]:

$$\text{TF-IDF}(t,d) = \text{TF}(t,d) \times \text{IDF}(t) \tag{1}$$

where Term Frequency (TF) measures how frequently a term t appears in a document d relative to the total number of terms in that document [33]:

$$\text{TF}(t,d) = \frac{\text{Frequency of } t \text{ in } d}{\text{Total terms in } d} \tag{2}$$

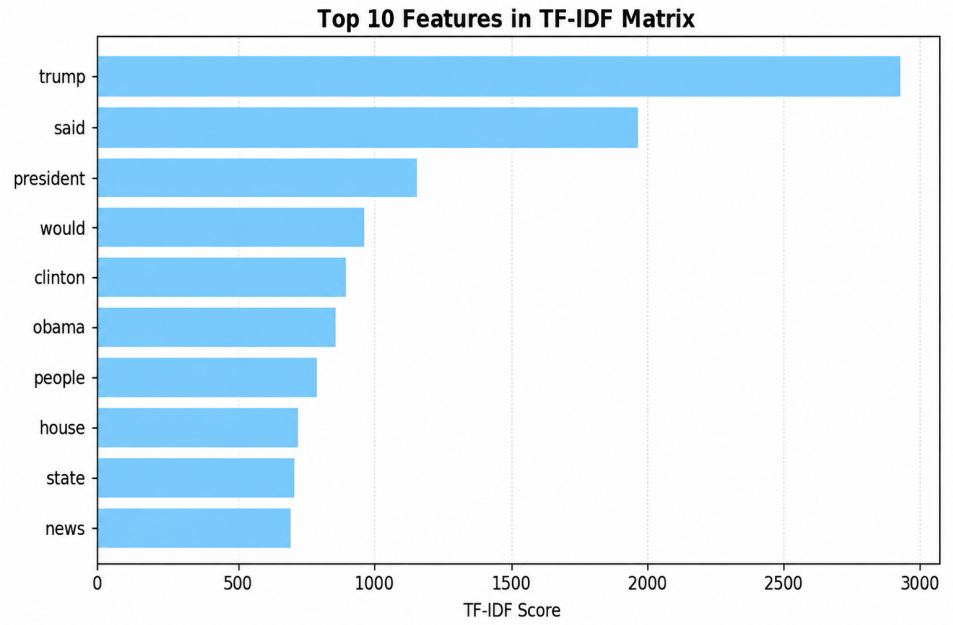

Figure 5. In this diagram represent the bar graph that illustrates the count of extracted features using TF-IDF

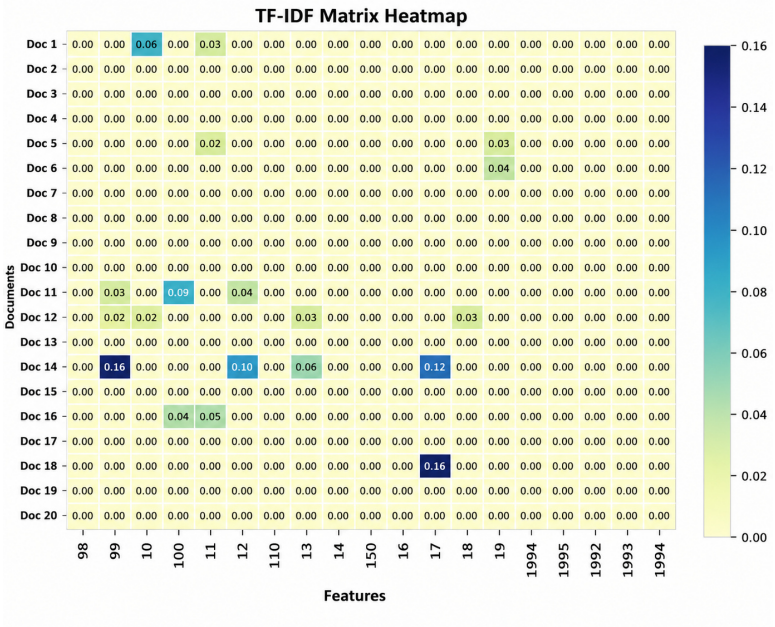

Figure 6. In this diagram represents the heatmap of extracted features using TF-IDF

4.4. Data Splitting

After performing feature extraction using the TF-IDF method, the next step involved dividing the dataset into two subsets: the training set and the test set. This division is essential for evaluating the model's ability to generalize to new, unseen data, helping to prevent overfitting, where the model may perform well on training data but fail to predict accurately on new data. In this study, the `train_test_split` function from the scikit-learn library was used to randomly split the dataset, ensuring that one subset is used for training and another for testing, thereby evaluating the model's generalization capability. The dataset was divided into two parts, with 80% of the data used for training the model and the remaining 20% reserved for testing purposes. Setting the `random_state=42` ensures that the data split remains the same each time the code is executed, providing consistency and reproducibility. This step is vital for assessing the model's performance on unseen data, providing a reliable indication of its generalization ability. The division of data between training and testing sets is illustrated in Figure 7, which shows the proportion of data used for each.

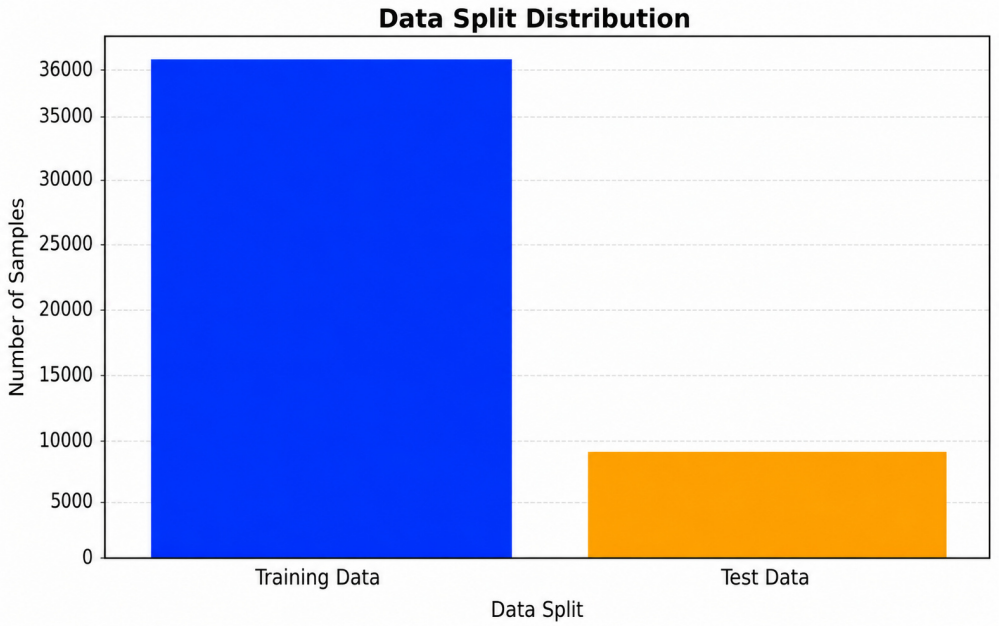

Figure 7. This diagram represents the bar graph of data splitting into two sets one is the training set and other is the testing set.

4.5. Feature Selection Using Particle Swarm Optimization and Genetic Algorithm

In this step, a total of 5000 extracted features were selected by utilizing a hybrid approach that integrates two powerful evolutionary algorithms: Particle Swarm Optimization and Genetic Algorithm. The reason for using this hybrid approach is to ensure that only the most relevant and significant features are chosen, thereby improving the model's ability to make accurate predictions. By combining the strengths of both algorithms, this method aims to identify the optimal subset of features that can contribute most effectively to the model's performance. The process began by performing feature selection using PSO, which helped identify a set of important features. Following this, the Genetic Algorithm was applied to further refine and select additional relevant features. Once the features selected by both algorithms were identified, they were combined into a unified set. This combined set of features was then used as input for a classifier algorithm to build the model. The combination of these carefully selected features ensures that the model is based on the most pertinent information, which is key for improving its overall performance. A detailed explanation of the entire process is provided below.

4.5.1. Particle Swarm Optimization

Particle Swarm Optimization is an optimization method that draws inspiration from the collective social behaviors observed in nature, such as the coordinated movements of bird flocks or fish schools. Introduced by James Kennedy and Russell Eberhart in 1995, PSO models the way individuals within a group share information and adapt their actions collaboratively to reach a shared objective. In PSO, individual particles adjust their movements based on both their own experience and the experience of other particles in the swarm, making it an extension of evolutionary algorithms [22]. Particle Swarm Optimization has some attractive features: it is easy to implement, doesn't require gradient information, and can be applied to a variety of optimization problems, including neural network training and function minimization [23]. PSO is highly effective in solving problems where the goal is to find optimal or near-optimal solutions within a large search space. In PSO, each particle represents a potential solution, and particles collaborate to explore the solution space. In this study, PSO is applied to feature selection for fake news detection. The objective is to improve the performance of a Logistic Regression model by selecting the most relevant features. Logistic Regression is chosen because it is a simple yet powerful classification algorithm, widely used for binary classification tasks such as distinguishing between fake and real news. Furthermore, using Logistic Regression during feature selection ensures that the features selected by PSO remain consistent across multiple runs. This consistency helps avoid fluctuations in feature selection, providing a more stable and reliable set of features for the model. PSO optimizes the selection of features, enhancing model performance, reducing overfitting, and improving generalization. In this study, the swarm size is set to 10 particles, ensuring the feature selection process is manageable and effective in exploring the feature space.

4.5.1.1. Steps in Particle Swarm Optimization

- Initialization: A population of 10 particles is randomly generated, where each particle is represented as a binary string. Each binary value (1 or 0) indicates whether a specific feature is included or excluded.
- Fitness Function Evaluation: The fitness value of each particle is computed by training a logistic regression model using the features selected by that particle. The accuracy of the model is used as the fitness score. A penalty is applied when no features are selected, ensuring the algorithm identifies meaningful subsets.
- Update Velocity and Positions: Particles update their velocity and position by considering three components: inertia (previous velocity), cognition (personal best solution), and social influence (global best solution). These updates allow particles to explore and converge toward better solutions iteratively.
- Optimization Over Iterations: The algorithm is run for 5 generations, with particles continuously refining their positions to

maximize fitness. The global best particle improves with each generation as the swarm converges toward optimal solutions.

- Selection of Features: At the end of the optimization process, the best-performing particle represents the final subset of features. In this case, PSO selected 1,030 features, as shown in Figure 8.

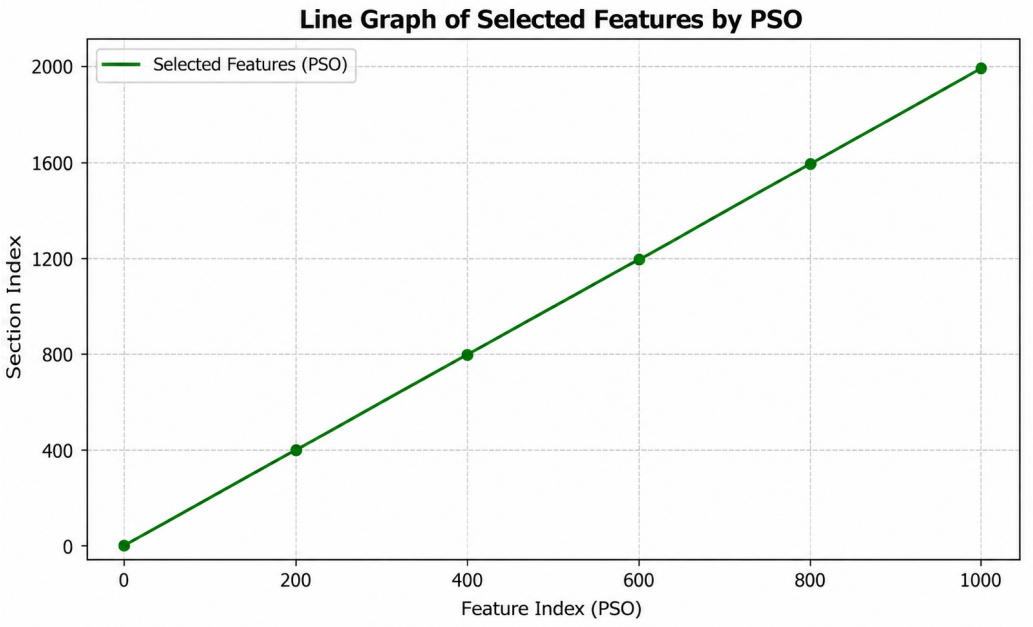

Figure 8. Presents a line graph of, the number of selected features using particle swarm optimization

4.5.2. Genetic Algorithm

The genetic algorithm is a prominent evolutionary algorithm, and John Holland, along with his students, worked on developing this algorithm in the 1960s, 1970s, and 1980s. By the mid-1980s, this algorithm began to be recognized by other research communities such as machine learning and operations research. It may not be a coincidence that the explosion of research into genetic algorithms coincided with the surge in research on artificial neural networks. This algorithm is inspired by biological systems as a computational model [25]. For instance, if binary strings represent real numbers in a GA for function optimization, operations like crossover, mutation, and selection are applied to these bit strings. To evaluate the fitness of a chromosome, the bit string is decoded into a real number or vector [24]. Genetic Algorithm is a computational technique inspired by the natural evolution process, where solutions evolve through selection, crossover, and mutation. It is particularly effective for optimizing complex problems by exploring large search spaces. In this study, GA is utilized for feature selection in fake news detection. The primary goal is to identify the most significant features that contribute to the effectiveness of a Logistic Regression model. Logistic Regression is chosen for its ability to efficiently handle binary classification tasks, such as differentiating between fake and real news. The inclusion of Logistic Regression in the feature selection process ensures that the selected features are consistent and reliable. By using Logistic Regression during each iteration of GA, the feature set remains stable, which minimizes the risk of irrelevant or fluctuating features affecting the model's performance. The GA proceeds by evolving a population of solutions over several generations, applying crossover and mutation operations to explore various feature subsets. The fitness of these subsets is evaluated based on their performance in the Logistic Regression model, and the best-performing feature subsets are retained. This process guarantees that the final selected features are both optimal and relevant, ultimately improving the accuracy of the fake news detection model.

4.5.2.1. Steps in Genetic Algorithm

- Population Initialization: The initial population comprises 10 binary strings, each representing a candidate solution. Similar to PSO, each bit indicates whether a feature is included (1) or excluded (0).
- Fitness Evaluation: Fitness is calculated by training a logistic regression model on the selected features. The model's performance on the dataset determines the fitness value. Solutions with no selected features are penalized to avoid invalid subsets.
- Selection, Crossover, and Mutation:
  - Selection: The tournament selection method is used to choose parent solutions based on their fitness.
  - Crossover: Parents are combined with a crossover probability of 0.7 to produce offspring.
  - Mutation: Random mutations with a probability of 0.2 introduce diversity by flipping bits in the offspring.
- Generational Evolution: GA operates over 5 generations, where new offspring replace the current population. The best-performing solutions are carried forward to maintain progress toward better feature subsets.
- Feature Subset: After 5 iterations, GA identifies the best subset of features, selecting 2,498 features in total. These results are illustrated in Figure 9.

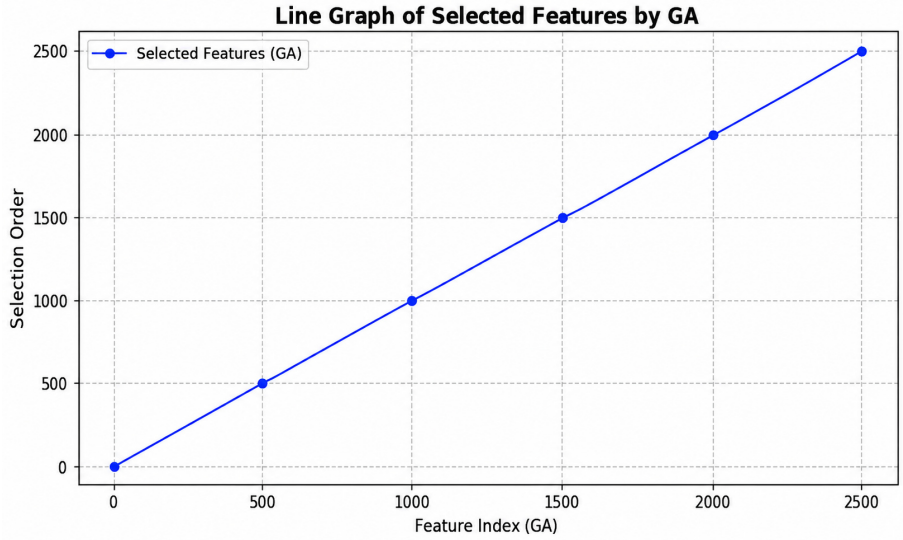

Figure 9. Presents a line graph, illustrating the number of selected features using genetic algorithm

4.5.3. Combined Feature Selection Using Particle Swarm Optimization and Genetic Algorithm

The features selected by both PSO and GA are merged to form a comprehensive and optimized set of features. The hybrid approach leverages the strengths of both algorithms, combining the exploration capabilities of PSO with the refinement capabilities of GA. The final feature set, consisting of 3,010 features, is visualized in Figure 10. This combined set represents the most relevant features identified through the collaborative efforts of both PSO and GA, ensuring an efficient and accurate feature selection process for fake news detection.

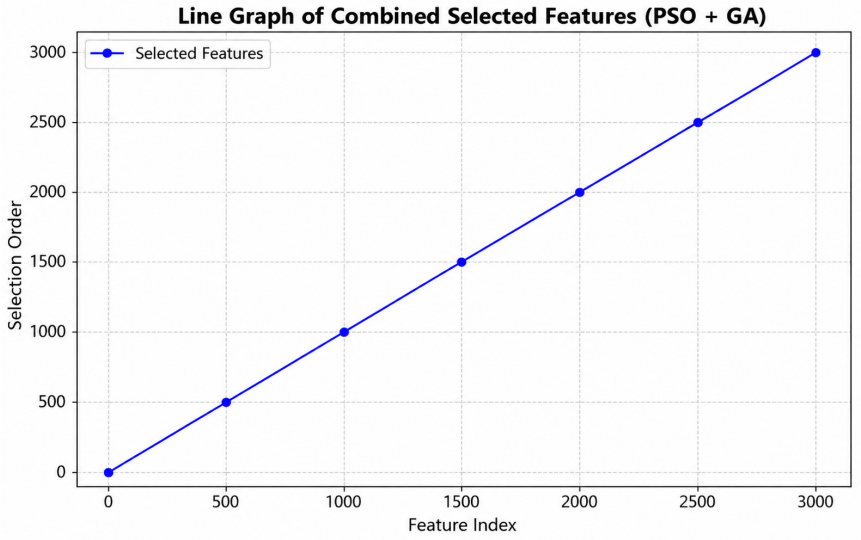

Figure 10. Presents a line graph, illustrating the number of selected features using combined particle swarm optimization and genetic algorithm approach(PSO+GA)

4.6. Classification Algorithm

This study utilized six distinct machine learning algorithms to assess their performance in addressing the problem. Below is a brief overview of each algorithm.

4.6.0.1. Logistic Regression

Logistic regression is a widely used technique for binary classification tasks, where the goal is to predict one of two possible outcomes. Unlike linear regression, which provides continuous numerical predictions, logistic regression focuses on estimating the likelihood of a particular category. It achieves this by applying a sigmoid function, which transforms the output into a probability value between 0 and 1. The model combines the input features linearly and uses a threshold, such as 0.5, to classify the outcome. Known for its simplicity and effectiveness, logistic regression is especially useful when there is a linear relationship between the predictors and the log-odds of the target variable.

4.6.0.2. Random Forest

Random Forest is a robust machine learning algorithm that can be applied to both classification and regression problems. It works by building multiple decision trees, each trained on different random subsets of the data and features, to ensure diversity. The algorithm aggregates the outputs of these trees to make a final prediction using majority voting for classification tasks and averaging for regression tasks. This ensemble approach improves accuracy and reduces the likelihood of overfitting compared to a single decision tree. Random Forest is well-suited for datasets with mixed feature types, can handle missing data, and is effective at capturing complex patterns in the data, making it a popular and reliable choice in various applications.

4.6.0.3. Gradient Boosting Classifier

The gradient boosting classifier is an advanced machine learning technique often used for classification problems. It builds a series of decision trees, where each new tree focuses on correcting the errors of the previous ones. By iteratively improving the model's predictions, it effectively reduces errors and enhances accuracy. This approach works by optimizing a loss function through a gradient descent process. It combines multiple weak models, such as shallow decision trees, into a single, strong predictive model. The method is highly flexible and can handle various data types, including numerical and categorical, while parameters like learning rate, the number of trees, and tree depth help manage model complexity and prevent overfitting. Although it can deliver high performance, gradient boosting often requires careful parameter tuning and can be computationally intensive. Frameworks like XGBoost, LightGBM, and CatBoost offer optimized implementations, making the process faster and more efficient for practical use.

4.6.0.4. Support Vector Machine

Support Vector Machine (SVM) is a supervised learning algorithm used for both classification and regression tasks. It operates by finding a hyperplane that separates data points into different classes with the maximum margin, ensuring a clear distinction between categories. The closest data points to this hyperplane, known as support vectors, play a key role in defining the boundary. SVM can handle both linear and non-linear problems. For non-linear cases, it uses kernel functions to map the data into a higher-dimensional space where separation becomes possible. Common kernels include linear, polynomial, and radial basis functions (RBF). The algorithm is effective for high-dimensional data and performs well when there is a clear margin of separation between classes. Although SVM is powerful, it may require careful tuning of parameters such as the kernel type and regularization strength to achieve the best results. It is computationally intensive for large datasets but excels in applications like text categorization, image classification, and bioinformatics, where precision in decision boundaries is essential.

4.7. Performance Metrics

After completing pre-processing, feature extraction, and feature selection, and successfully training the model, the final step involves evaluating its performance. This is achieved using standard metrics such as accuracy, precision, recall, F1-score, and the confusion matrix. These metrics provide a comprehensive understanding of the model's effectiveness and help identify areas for improvement. These metrics provide a clear understanding of its strengths and weaknesses. This study focuses on five key metrics commonly used to evaluate classification models: Accuracy, Precision, Recall, F1-Score, and Support, and also evaluates performance using a heatmap, line graph, and bar graph. Below is an explanation of each metric:

4.7.0.1. Accuracy

Accuracy is a key metric that measures the percentage of correctly classified samples about the total number of samples in a dataset. It offers an overall measure of the model's predictive capability. While accuracy is effective for datasets with balanced class distributions, it may be misleading in cases of class imbalance. For example, in a dataset where 90% of instances belong to one class, a model that predicts the majority class for all cases may achieve high accuracy, yet perform poorly for the minority class [34].

4.7.0.2. Precision

Precision calculates the proportion of true positive predictions relative to all instances predicted as positive by the model. It indicates the model's capability to minimize incorrect positive predictions. A True Positive (TP) is when the model correctly predicts a positive instance, whereas a False Positive (FP) is when a negative instance is mistakenly predicted as positive [35].

4.7.0.3. Recall

Also referred to as sensitivity, recall measures the proportion of actual positive instances that the model correctly identifies. It focuses on minimizing false negatives, ensuring that all relevant positive instances are captured [36].

4.7.0.4. F1-Score

The F1-Score represents the harmonic mean of precision and recall, offering a balanced measure of a model's performance. It is especially beneficial for imbalanced datasets as it accounts for both false positives and false negatives. This metric is particularly valuable when balancing precision and recall is critical [37].

4.7.0.5. Support

Support represents the number of actual instances for each class within the dataset. This metric highlights the class distribution, which is essential for contextualizing the other performance metrics.

4.7.0.6. Confusion Matrix

The confusion matrix provides a detailed visualization of a classification model's performance by comparing predicted values against actual values. It classifies outcomes into four categories:

- True Positive (TP): Correctly predicted positive cases.
- True Negative (TN): Correctly predicted negative cases.
- False Positive (FP): Incorrectly predicted positive cases.
- False Negative (FN): Incorrectly predicted negative cases.

	Predicted Positive	**Predicted Negative**
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	False Positive (FP)	True Negative (TN)

Table 1. Confusion Matrix: The confusion matrix summarizes a classification model's performance by displaying the counts of true positives (TP), false negatives (FN), false positives (FP), and true negatives (TN) for both predicted and actual classes

4.7.0.7. Heatmap

A heatmap is a graphical representation of data where varying colors depict individual values. This visualization technique is commonly used to show the intensity or magnitude of data points in a matrix or grid format. The colors in a heatmap typically range from light to dark or use a color gradient to represent numerical values, making it easier to identify patterns, trends, or correlations within the data. Heatmaps are widely applied in various fields like data analysis, machine learning, and statistics, as they provide a quick and intuitive way to visualize complex information.

5. Experimental Analysis and Results

This study utilizes the FakeNewsNet dataset, which includes the columns: title, text, date, label, and subject. For this research, the focus is on the title, text, subject, and label columns. Among these, the label column serves as the target variable, while the title, text, and subject columns were pre-processed to prepare them for further analysis. Following pre-processing, 5,000 features were extracted using the Term Frequency-Inverse Document Frequency (TF-IDF) method. A hybrid approach combining genetic algorithms and particle swarm optimization was employed to enhance the feature selection process. Both GA and PSO are powerful algorithms within the realm of evolutionary computation, known for their ability to optimize complex problems. The hybrid technique effectively selected the most relevant features, which were then used to train four different machine-learning algorithms. The performance of these machine learning models was thoroughly evaluated using standard metrics, including accuracy, precision, recall, and F1-score. This comprehensive evaluation ensured a detailed understanding of the strengths and weaknesses of each algorithm. The results of the experiments are presented in a detailed and structured manner through comprehensive tables and figures, as outlined below: In Figure 11 displays the confusion matrices for Logistic Regression and SVM, presented as heatmaps, providing a clearer understanding of their prediction performance. Figure 12 showcases the heatmap of the confusion matrices for Gradient Boosting and Random Forest, allowing for an immediate visual comparison of these two algorithms. Figure 13 compares the accuracy and precision of the four machine learning algorithms using a bar graph, effectively illustrating their relative strengths. Figure 14 continues this comparison by presenting recall and F1-score in a bar graph, further emphasizing the distinctions between the algorithms. Figure 15 consolidates all four metrics—accuracy, precision, recall, and F1-score—into a single, comparative bar graph, offering a holistic view of the performance across all algorithms. Figure 16 presents a heatmap that visualizes the performance of all four machine learning algorithms, clearly displaying their accuracy, precision, recall, and F1 score in an easily interpretable format. Finally

Classifier	Accuracy	Precision	F1-Score	Recall	Support
Logistic Regression	0.999	0.999	0.999	0.999	8980
Random Forest (RF)	1.00	1.00	1.00	1.00	8980
Gradient Boosting Classifier (GB)	1.00	1.00	1.00	1.00	8980
Support Vector Machine (SVM)	1.00	1.00	1.00	1.00	8980

Table 2. Compares the performance of Four classifiers in accuracy, F1-score, recall, and support

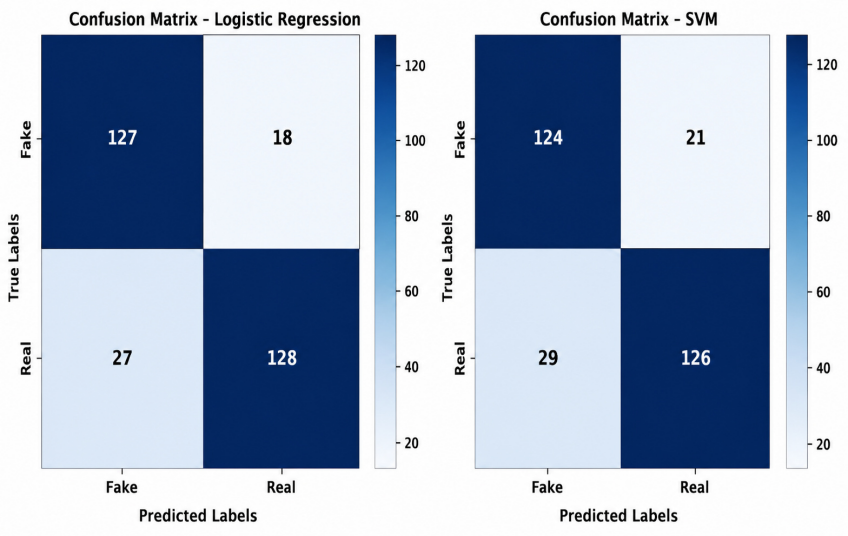

Figure 11. Presents a heatmap of the confusion matrix of logistic regression and support vector machine

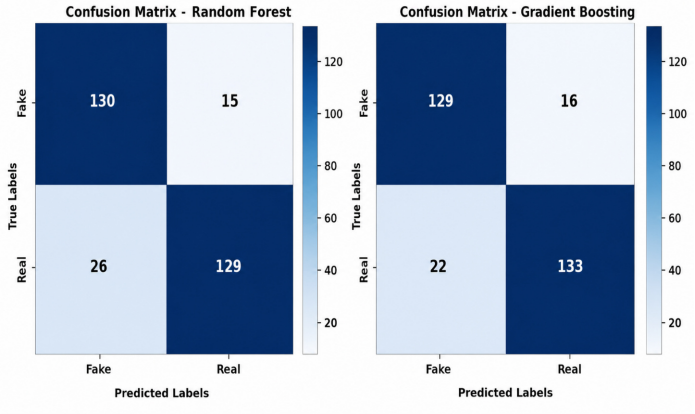

Figure 12. Presents a heatmap of the confusion matrix of random forest and gradient boosting classifier

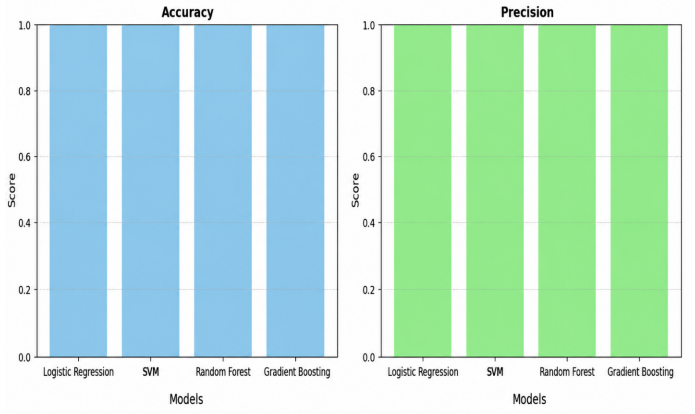

Figure 13. A comparison of four classifiers is presented, highlighting their performance in terms of accuracy, precision. The bar graph illustrates the accuracy and precision of various classification models, including Logistic Regression (LR), Random Forest (RF), Support Vector Machine (SVM), Gradient Boosting Classifier. The analysis reveals that while the accuracy and precision of all the algorithms is almost equal, the accuracy and precision of the Logistic Regression is notably lower than that of the other algorithms

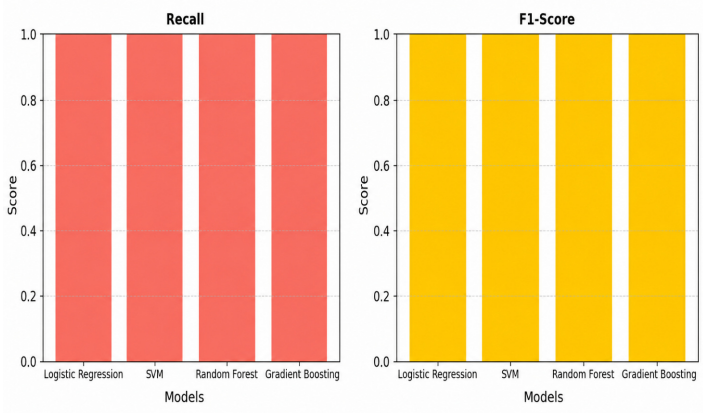

Figure 14. A comparison of four classifiers is presented, highlighting their performance in terms of recall and F1-Score. The bar graph illustrates the recall of various classification models, including Logistic Regression (LR), Random Forest (RF), Support Vector Machine (SVM), Gradient Boosting Classifier. The analysis indicates that the recall and F1-Score of all the algorithms are nearly equal; however, the recall and F1-Score for the Logistic Regression model is Lower than that of the other algorithms.

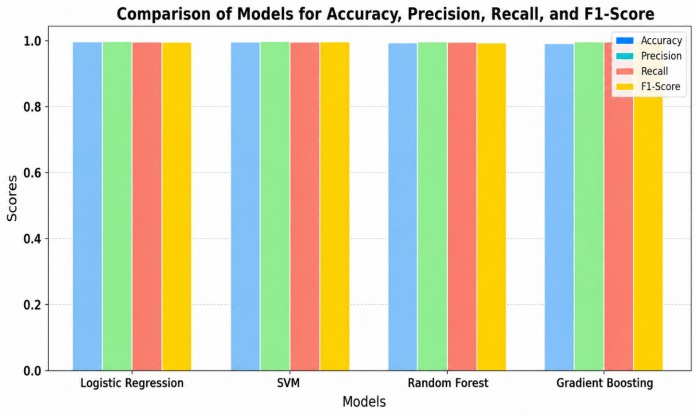

Figure 15. Four classifiers are compared, highlighting their performance in terms of accuracy, precision, recall, and F1-Score. The bar graph displays the performance of these four classification models, revealing that the Logistic Regression algorithm achieves the lowest performance compared to the other models

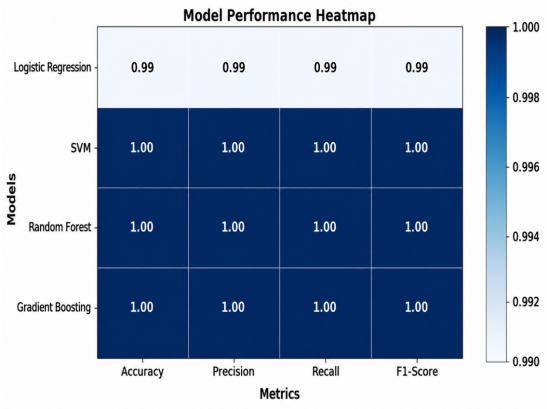

Figure 16. Confusion Matrix of Different Classification Algorithms

Figure 17 provides a line graph comparison of the four algorithms, offering an alternative perspective on their relative performance across different evaluation metrics. These enhanced visualizations and comparisons offer a deeper understanding of the machine learning algorithms' capabilities, highlighting the significant role of the hybrid feature selection technique in improving classification performance and its potential to address the challenges posed by fake news detection.

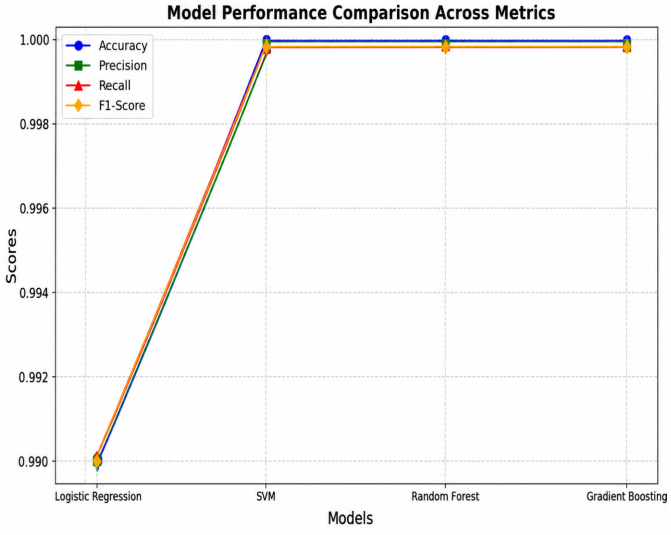

Figure 17. Line Graph of different classification algorithm

6. Conclusion and Future Scope

This study successfully developed a robust fake news detection model using a hybrid feature selection method that combined Genetic Algorithms and Particle Swarm Optimization. The research was conducted using the FakeNewsNet dataset, which comprises two datasets: one containing fake news and the other containing true news. These two datasets were merged to create a single dataset with 44,898 news articles and five columns. This dataset is publicly available on Kaggle. By applying this hybrid approach, the most relevant features were identified and used to train four different machine learning classifiers. The results demonstrated that three classifiers achieved a perfect accuracy of 100%, while Logistic Regression performed slightly lower with 99% accuracy. These findings emphasize the effectiveness of combining evolutionary algorithms for feature selection in improving the performance of fake news detection systems. Although this research primarily concentrated on textual data, there is significant potential for extending this approach to handle multimedia content, such as images, videos, and audio, frequently involved in spreading misinformation. Incorporating deep learning models for image and video analysis, such as Convolutional Neural Networks, could enhance the model's ability to analyze visual and audio cues alongside textual data. By exploring integrating these multimodal data types, future work could lead to a more comprehensive detection system capable of identifying fake news across various formats.

ACKNOWLEDGEMENTS

We would like to express our appreciation to the University for providing the necessary resources and support for this research. First and foremost, I would like to express my heartfelt gratitude to my family and to God for their unwavering support and guidance. I am deeply thankful for their encouragement and belief in me. I would also like to extend my sincere thanks to my supervisor for their invaluable guidance and support in this research. Their expertise and mentorship have been instrumental in shaping this work.

REFERENCES

[1] https://github.com/KaiDMML/FakeNewsNet.

[2] 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.

[3] Kuhn, M., & Johnson, K. (2013). Applied predictive modeling. Springer Science & Business Media. https://doi.org/10.1007/978-1-4614-6849-3.

[4] Guyon, I., & Elisseeff, A. (2003). An introduction to feature selection. Journal of Machine Learning Research, 3, 1157-1182. https://doi.org/10.1162/15324430332275316.

[5] Pradnya A. Vikhar (2016). Evolutionary Algorithms: A Critical Review and its Future Prospects. International Conference on Global Trends in Signal Processing, Information Computing and Communication, Volume 4, issue 9, pp. 758-766. https://doi.org/10.1109/ICGTSPICC.2016.7955308.

[6] Amrita Chakraborty and Arpan Kumar Kar (2017). Swarm Intelligence: A Review of Algorithms. ResearchGate. https://doi.org/10.1007/978-3-319-50920-4_19.

[7] Back, T. (1996). Evolutionary algorithms in theory and practice: Evolution strategies, evolutionary programming, genetic algorithms. Oxford University Press. https://doi.org/10.1002/widm.1124.

[8] Holland, J. H. (1975). Adaptation in natural and artificial systems. University of Michigan Press. https://doi.org/10.1145/1216504.1216510.

[9] Mitchell, M. (1996). An introduction to genetic algorithms. MIT Press.

[10] Eiben, A. E., et al. (2003). Evolutionary algorithms: A critical review. Natural Computing, 2(3), 135-148. https://doi.org/10.1007/978-3-319-45403-0_6.

[11] Kennedy, J., & Eberhart, R. C. (1995). Particle swarm optimization. In Proceedings of IEEE International Conference on Neural Networks (Vol. 4, pp. 1942-1948). https://doi.org/10.1109/ICNN.1995.488968.

[12] Zhao, Z., et al. (2018). A hybrid GA-PSO optimization algorithm for feature selection in high-dimensional data. IEEE Access, 6, 51132-51141.

[13] Chouhan, S. S., et al. (2018). Hybrid GA-PSO optimization for feature selection in large-scale classification problems. Soft Computing, 22(12), 4231-4243.

[14] Pudil, P., et al. (1994). Floating search methods in feature selection. Pattern Recognition Letters. https://doi.org/10.1016/0167-8655(94)90127-9.

[15] Jiang, L., et al. (2020). PSO-based feature selection for fake news detection. Neural Computing and Applications.

[16] Zhou, L., et al. (2019). Genetic algorithm-based feature selection for fake news detection. Journal of Artificial Intelligence Research.

[17] Zhou, L., et al. (2020). Hybrid GA-PSO feature selection for fake news detection. Expert Systems with Applications.

[18] Deb, K., et al. (2018). Hybrid GA-PSO feature selection for image classification. International Journal of Computer Vision.

[19] Ruchansky, N., et al. (2017). Csi: A hybrid deep learning approach for fake news detection. Proceedings of the 2017 ACM Conference on Information and Knowledge Management. https://doi.org/10.1145/3132847.3132877.

[20] Ahmed, H. F., et al. (2018). A survey on fake news detection techniques. IEEE Access.

[21] Niood Mohammed AlShariah, Abdul Khader Jilani Saudagar (2019). Detecting Fake Images on Social Media Using Machine Learning. (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 10, No. 12. https://doi.org/10.14569/IJCSA.2019.0101224.

[22] A. A. A. Esmin, G. Lambert-Torres, G. B. Alvarenga (2006). Hybrid Evolutionary Algorithm Based on PSO and GA mutation, IEEE, Proceedings of the Sixth International Conference on Hybrid Intelligent Systems (HIS'06). http://dx.doi.org/10.1109/HIS.2006.264940.

[23] Y. Shi and R. C. Eberhart (1998). Parameter Selection in Particle Swarm Optimization. Evolutionary Programming VII, Lecture Notes in Computer Science 1447, 591–600. Springer. http://dx.doi.org/10.1007/BFb0040810.

[24] Xin Yao (1997). Global Optimisation by Evolutionary Algorithms, Computational Intelligence Group, School of Computer Science, University College, the University of New South Wales, Australian Defence Force Academy, Canberra ACT, Australia 2600, pg. 282.

[25] Darrell Whitley (2001). An overview of evolutionary algorithms: practical issues and common pitfalls, Elsevier, Information and Software Technology, Vol. 43, Issue. 14, pp. 817-831. http://dx.doi.org/10.1016/S0950-5849(01)00188-4.

[26] Y. Li, & S. Cha (2019). Face Recognition System. arXiv preprint arXiv:1901.02. https://doi.org/10.48550/arXiv.1901.02452.

[27] Rohit Kaliyar, Anurag Goswami, Pratik Narang (2019). Multiclass Fake News Detection Using Ensemble Machine Learning, December. http://dx.doi.org/10.1109/IACC48062.2019.8971579.

[28] Hnin Ei Wyne, Zar Zar Wint (2019). Content-Based Fake News Detection Using N-Gram Model, December. http://dx.doi.org/10.1145/3366030.3366116.

[29] Anastasia Giachanou, Guobiao Zhang, Paolo Rosso (2020). Multimodal Multi-image Fake News Detection. ResearchGate. http://dx.doi.org/10.1109/DSAA49011.2020.00091.

[30] Shen Hoe Kong, Li Mei Tan, Keng Hoon Gan, Nur-Hana Samsudin (2020). Fake news detection using Deep Learning. ResearchGate. https://doi.org/10.13140/RG.2.2.31151.

[31] Abdullah Yanher, Amer Alhariri (2021). Detection of Covid-19 Fake News Text Using Random Forest and Decision Tree Classifiers. https://doi.org/10.5281/zenodo.4427205.

[32] Z Khanam, B N Alwasel, H Sirafi, Mamoon Rashid (2021). Fake News Detection Using Machine Learning Approaches. IOP Conference Series: Materials Science & Engineering, 1099(1):022040. https://doi.org/10.1088/1757-899X/1099/1/012040.

[33] Ahmed Hasim Jawad Almarashy, Mohammad-Reza Feizi-Derakhshi, Pedram Salehpour (2023). Enhancing Fake News Detection by Multi-Feature Classification. IEEE Access. https://doi.org/10.1109/ACCESS.2023.3339621.

[34] H. Saleh, A. Alharbi, and S. H. Alsamhi (2021). OPCNN-FAKE: Optimized convolutional neural network for fake news detection. IEEE Access, vol. 9, pp. 129471–129489. https://doi.org/10.1109/ACCESS.2021.3112806.

[35] R. K. Kaliyar, A. Goswami, and P. Narang (2021). Fake BERT: Fake news detection in social media with a BERT-based deep learning approach. Multimedia Tools Appl., vol. 80, no. 8, pp. 11765–11788. https://doi.org/10.1109/ACCESS.2023.3339621.

[36] S. Kumar, R. Asthana, S. Upadhyay, N. Upreti, and M. Akbar (2020). Fake news detection using deep learning models: A novel approach. Trans. Emerg. Telecommun. Technol., vol. 31, no. 2, p. e3767. https://doi.org/10.1002/ett.3767.

[37] R. K. Kaliyar, A. Goswami, P. Narang, and S. Sinha (2020). FNDNet—A deep convolutional neural network for fake news detection. Cognit. Syst. Res., vol. 61, pp. 32–44.

[38] Himank Gupta, Mohd. Saalim Jamal, Sreekanth Madisetty, Maunendra Sankar De Sarkar (2018). A Framework for Real-Time Spam Detection in Twitter. https://doi.org/10.1109/COMSNETS.2018.8328222.

[39] M.F. Mridha, Ashifa Jannat Keya, MD. Abdul Hamid et al. (2022). A Comprehensive Review of Fake News Detection with Deep Learning. IEEE Access. https://doi.org/10.1109/ACCESS.2021.3129329.

[40] Bodunde Akinyemi, Oluwakemi Adewusi, Adedoyin Oyebade (2020). An Improved Classification Model for Fake News Detection in Social Media. International Journal of Information Technology and Computer Science, 12(1):34-43. https://doi.org/10.5815/ijitcs.2020.01.05.