EUROPEAN JOURNAL OF PURE AND APPLIED MATHEMATICS
2025, Vol. 18, Issue 2, Article Number 6087
ISSN 1307-5543 – ejpam.com
Published by New York Business Global

Towards Accurate Fake News Detection: Evaluating Machine Learning Approaches and Feature Selection Strategies

Mutaz A. B. Al-Tarawneh[1,*], Ashraf Al-Khresheh[2], Omar Al-irr[1], Ajla Kulaglic[1], Kassem Danach[3], Hassan Kanj[1], Ghayth AlMahadin[4]

[1] *College of Engineering and Technology, American University of the Middle East, Egaila 54200, Kuwait*
[2] *Computer Science Department, Tafila Technical University,Tafila, Jordan*
[3] *Basic and Applied Sciences Research Center, Al Maaref University, Beirut, Lebanon*
[4] *Data Science Department, Mutah University, Karak, 61710, Jordan*

Abstract. The rapid spread of fake news in the digital age poses significant challenges, necessitating effective detection methods. This study presents a comprehensive evaluation of various ensemble and machine learning classifiers, combined with different feature selection techniques, to improve the accuracy and reliability of the detection of fake news. Using the TruthSeeker dataset, this research examines feature selection methods such as Recursive Feature Elimination (RFE), SelectKBest, Principal Component Analysis (PCA), and Genetic Algorithms (GA), analyzing their impact on model performance. Key metrics such as accuracy, precision, recall, F1 score, and AUC-ROC were used to assess the effectiveness of each classifier. The results reveal that ensemble methods, particularly Random Forest (RF) and Gradient Boosting, demonstrate superior performance, achieving high accuracy and AUC-ROC scores. Moreover, feature selection techniques like RFE and SelectKBest significantly improve model outcomes by optimizing the feature set, while PCA is less effective in this context. This study highlights the importance of integrating robust classifiers with optimal feature selection methods to improve the efficacy of fake news detection systems.

2020 Mathematics Subject Classifications: 68T01

Key Words and Phrases: Fake news detection, Machine learning, Feature selection, Truthseeker

*Corresponding author.
DOI: https://doi.org/10.29020/nybg.ejpam.v18i2.6087

Email addresses: mutaz.al-tarawneh@aum.edu.kw (M. Al-Tarawneh), khashraf@ttu.edu.jo (A. Al-Khresheh), omar.alirr@aum.edu.kw (O. Alirr), ajla.kulaglic@aum.edu.kw (A. Kulaglic), kassem.danach@iul.edu.lb (K. Danach), hassan.kanj@aum.edu.kw (H. Kanj), ghayth.mahadin@mutah.edu.jo (G. AlMahadin)

1. Introduction

The advent of the digital age has transformed the way information is accessed and disseminated, leading to unprecedented opportunities for knowledge sharing. However, this increased accessibility has also facilitated the widespread circulation of misinformation and fake news. Fake news, characterized as deliberately false or misleading content presented in the guise of authentic news, has emerged as a critical challenge with far-reaching consequences. It influences public perception, shapes political decisions, and can even lead to social instability. The difficulty of distinguishing genuine news from fabricated stories is exacerbated by the rapid and extensive distribution of information through social media and digital platforms [1].

The repercussions of fake news extend into the physical world, eroding trust in traditional media and impacting real-world events. For example, the 2016 US presidential election saw a proliferation of misleading stories that may have affected voter behavior and the final outcome [2]. Similarly, during the 2019 Indian general election, targeted misinformation campaigns spread false narratives and incited division among voters [3]. The COVID-19 pandemic further highlighted the dangers of false news, with widespread misinformation about the virus, treatments, and vaccines creating significant public health risks [4]. These instances underscore the pressing need for robust strategies to identify and mitigate fake news.

Machine learning (ML) has become a cornerstone in addressing this challenge, offering advanced methods to identify and classify fake news. By analyzing extensive datasets with sophisticated algorithms, ML models can detect patterns and attributes indicative of misinformation [5]. The fake news detection process typically involves several stages: data collection, feature extraction, model training, and evaluation. Among these, feature selection plays a pivotal role in pinpointing the most relevant attributes in the dataset, thereby enhancing model accuracy and efficiency.

Feature selection is crucial to improve the performance of the ML model, as it reduces complexity and increases interpretability [6]. Several techniques have been developed for this purpose. Recursive Feature Elimination (RFE) incrementally removes less relevant features based on model weights to refine the set of features [7]. SelectKBest applies statistical tests to identify the key features that most closely correlated with the target variable [8]. Principal Component Analysis (PCA) transforms existing features into a smaller set of components, retaining maximum variance [9]. Furthermore, Genetic Algorithms (GAs) offer a global optimization approach by mimicking natural selection processes to identify the optimal set of characteristics in complex search spaces [10].

This research presents a comprehensive evaluation of machine learning methods for detecting fake news. Explores the performance of several individual classifiers, including Decision Trees (DT), Naïve Bayes (NB), Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Random Forest (RF), and Multilayer Perceptron (MLP), along with ensemble approaches like Bagging, Boosting, and Stacking. An emphasis is placed on the influence of various feature selection methods on model effectiveness. A robust evaluation framework that employs metrics such as accuracy, precision, recall, F1 score, and

AUC-ROC ensures a thorough analysis of the model capabilities.

A key asset in this research is the TruthSeeker dataset, which encompasses a decade worth of news articles, tweets, and social media content. This diverse dataset supports a detailed analysis by providing textual, lexical, and metadata features. Inclusion of metadata, such as user engagement statistics, allows the detection of subtle patterns that textual analysis alone may miss. The temporal and content diversity of the dataset ensures that the models are evaluated across a wide range of real-world scenarios.

By systematically comparing individual classifiers and ensemble methods alongside various feature selection techniques, this study sheds light on the optimal strategies for effective fake news detection. The contributions of this work include the following.

- **Comprehensive Evaluation:** A detailed assessment of multiple machine learning models for the detection of fake news using various performance metrics such as accuracy, precision, recall, F1 score and AUC-ROC.
- **Impactful Feature Selection:** Analysis of the effects of feature selection methods, including RFE, SelectKBest, PCA, and GA, on model performance.
- **Rich Dataset Utilization:** Utilization of the TruthSeeker dataset, offering diverse and temporally extensive data to ensure robust model evaluation.
- **Holistic Analysis:** Examination of textual, lexical, and metadata features to capture nuanced patterns in the detection of fake news.
- **Comparative Analysis:** Insights into the performance of individual classifiers versus ensemble methods to identify optimal detection strategies.

The remainder of this paper is structured as follows. Section 2 reviews related work, Section 3 details the methodology, including data collection, machine learning techniques, feature selection, and evaluation. Section 4 discusses the results and findings, and Section 5 concludes the study with key findings.

2. Literature Review

Fake news detection has emerged as a critical research area due to the pervasive spread of misinformation across digital platforms. In recent years, several studies have explored ML techniques to tackle this pressing issue. This review of the literature discusses notable work in the field, highlighting their methodologies and comparing them with current research.

The study in [11] presented BiL-FaND, an ensemble-based system for detecting fake news in English and Urdu. Combining multilingual BERT, long-short-term memory (LSTM) models, and a ResNet-101 with GRU for multimedia analysis, the system achieved an overall notable accuracy of 92.07%. The approach demonstrated strong performance in linguistic analysis, pattern recognition, and multimedia content evaluation. However, the study does not explore feature selection methods.

The authors of [12] developed a robust fake news detection approach using WELFake, FakeNewsNet, and FakeNewsPrediction datasets. FastText embeddings were combined with ML and deep learning (DL) methods, including a hybrid CNN-LSTM model that achieved high performance (accuracy and F1 scores of 99%, and 97%). Transformer-based models (BERT, XLNet, RoBERTa) were also optimized for superior syntactic management. Explainable AI techniques (LIME, LDA) provided insight into model decisions. However, the study does not explore feature selection methods.

The work in [13] addressed the spread of COVID-19 misinformation by developing a Convolutional Neural Network (CNN)-based deep learning model using word embeddings. The optimal CNN architecture was determined through grid search. The effectiveness of the model was evaluated against various state-of-the-art ML algorithms, with CNN achieving the highest performance: 96.19% mean accuracy, 95% mean F1 score, and 0.985 AUC. However, they did not explore feature selection methods.

The study in [14] examined the impact of fake news on society and presented a comparative analysis of ML methods to detect fake news. The models evaluated included NB, DT classifier, RF, and logistic regression (LR). The results indicated that the DT classifier achieved the highest accuracy at 99.56%, followed by RF (99. 35%), LR (98. 91%) and NB (94. 89%). Recall results were also better for DT and RF compared to LR and NB. In particular, the study focused on model comparison rather than exploring feature selection methods.

The research proposed in [15] introduced an innovative approach to detecting fake news by analyzing semantic discrepancies between the titles and content of the articles. Using feature selection, the contents of the articles are summarized, and the vector distances (Cosine similarity, Jaccard distance, Euclidean distance) between the titles and the contents are calculated. These distances are combined as features to train the ML and DL models, achieving almost 99.9% accuracy in three data sets. This method demonstrates the effectiveness of exploiting semantic differences and underscores the importance of feature selection in enhancing detection performance.

The approach of [16] addressed the rapid spread of fake news by investigating style-based detection using text analysis with natural language processing techniques. Various models were built using word representations (TF-IDF, word2Vec) and ML (e.g., KNN, NB, LR) and DL (e.g., LSTM) methods on the ISOT dataset. Performance was evaluated using accuracy, precision, recall, and the F1 score, with the LSTM model achieving the highest accuracy at 99.2%. Although the study focused on improving word representations and classification methods, it did not investigate feature selection techniques, which are also important to optimize model performance.

The study in [17] explored various ML algorithms to identify fake news, with the aim of determining which algorithm achieved the highest precision. The study found that the DT algorithm performed the best, achieving an accuracy of 99.97% compared to other methods. However, the research did not investigate feature selection techniques, which are also crucial to enhancing model performance.

The study in [18] explored various ML models to classify the veracity of information and detect fake news on social networks. The analysis used a dataset of approximately 40,000

items (20,000 fake and 20,000 real news) and used a variety of models, including SVM, LR, CatBoost, XGBoost, multinomial NB, and RF. Performance was evaluated using metrics such as accuracy, precision, recall, F1 score, and more. The deep Auto_ViML model achieved the highest accuracy, precision, recall and F1 score at 99%, while the hybrid learning model had the best false rejection rate at 71%. The SVM was noted for its computational efficiency, taking only 0.245 ms for computation. However, the study did not investigate feature selection techniques, which are also crucial to enhancing model performance.

The authors of [19] proposed a method that uses ML classification to distinguish between genuine and fake content. The approach used natural language processing techniques, including the TF-IDF and Word2Vec features, which were optimized using the RFE algorithm for feature selection. Training and testing were performed on the Kaggle 'Fake News Detection' dataset, and the results showed that the boosting ensemble methods outperformed other techniques. The study highlighted the effectiveness of feature selection through RFE, but did not explore other feature selection methods.

The study in [20] aimed to build a model to detect fake news by analyzing the characteristics of the text. TF-IDF technology was used to convert words into features and identify the highest-ranking features to distinguish between real and fake news. The study then adapted ML techniques including LR, DT, gradient boosting and RF, with DT and Gradient Boosting achieving the highest precision at 99.4% and 99.49%, respectively. Although the study focused on feature extraction through TF-IDF, it did not explore other feature selection methods, which could have further enhanced model performance.

The research presented in [21] evaluated the effectiveness of three ML algorithms-Multinomial NB, Passive Aggressive Classifier, and LR-to distinguish between fake and genuine news articles. Using a balanced dataset, the study processed and vectorized the text with TF-IDF and assessed algorithms based on precision, recall and F1 score. The Passive Aggressive Classifier achieved the highest performance with a precision and recall of 0.99, while LR had an accuracy of 0.98 and Multinomial NB exhibited a robust recall of 100% but lower precision at 91%, resulting in an accuracy of 95%. The study did not explore advanced feature selection methods beyond TF-IDF, which could have further improved the models' performance.

The paper in [22] addressed the challenge of fake news detection by proposing the use of various AI and ML methods, combining both individual and ensemble approaches for automated news classification. The study explored different textual properties to distinguish fake news from real news and applied multiple ML algorithms to a real-world dataset for this classification task. The performance of these models was evaluated on the basis of their accuracy in identifying fake news. However, the study did not discuss advanced feature selection techniques beyond basic textual properties.

The research in [23] addressed the challenge of fake news detection in Pakistan by creating a comprehensive dataset from multiple fact-checked news APIs and evaluating it using a range of ML and DL techniques. The study applied five ML algorithms: NB, KNN, LR, SVM, and DT, as well as two DL models: CNN and LSTM using GloVe and BERT embeddings for text representation. The models were assessed based on precision, F1

score, accuracy, and recall, with the LSTM model using GloVe embeddings achieving the highest performance. However, the study did not investigate feature selection methods but focused on different embeddings and algorithms for classification. In addition, the research included an analysis of misclassified samples by comparing them with human judgments.

The study in [24] addressed the challenge of fake news detection by developing a model that incorporates user characteristics, news content, and social network features based on social capital. The XGBoost model was used to estimate the importance of the features and identify key variables for the detection of fake news. Several ML algorithms, including SVM, RF, LR, Classification and Regression Trees (CART) and Neural Networks (NN), were applied to classify news articles. A cross-validation step was performed to generalize the models and prevent overfitting or underfitting. The RF model achieved the highest prediction rate of approximately 94%, while the NN model had the lowest performance of about 92.1%. The study did not explore traditional feature selection methods but focused on derived features based on social capital and estimated their importance to detect fake news. The work in [25] provided valuable information on the impact of different word embedding techniques on the performance of machine learning and deep learning models for the detection of fake news. Although that study focused on embedding-based feature representations and compared individual classifiers and CNN-based architectures, it did not consider ensemble learning techniques or feature selection strategies beyond embeddings. In contrast, the current study adopts a different research direction by conducting a comprehensive evaluation of traditional and ensemble classifiers in combination with multiple feature selection techniques, including Recursive Feature Elimination (RFE), SelectKBest, Principal Component Analysis (PCA), and Genetic Algorithms (GA). This approach enables a deeper investigation into the relevance and interpretability of features and offers comparative information on how various selection methods influence classification performance, areas not covered in the previous work.

In contrast to these studies, this research presents several distinctive strengths. Firstly, this study employs a diverse set of individual classifiers, including DT, NB, SVM, KNN, RF, and MLP, along with ensemble methods such as Bagging, Boosting, and Stacking. This comprehensive approach ensures a robust evaluation of various algorithms for the detection of fake news. In addition, the research systematically investigates the impact of various feature selection methods, including RFE, SelectKBest, PCA, and GA, on model performance. This detailed examination of feature selection techniques provides valuable insight into how different methods influence the effectiveness of fake news detection models. Furthermore, the study uses the TruthSeeker dataset, which offers a rich and varied source of data comprising textual, lexical, and metadata features. Spanning more than a decade, this extensive dataset ensures that the models are tested on a wide range of real-world scenarios. The inclusion of metadata features, such as user behavior and engagement metrics, allows for a more nuanced analysis of fake news, capturing patterns that may not be evident from textual analysis alone. This holistic approach improves the accuracy and robustness of fake news detection models, which separates this study from others that focus primarily on advanced model architectures or specific feature sets. In summary,

while recent studies have made significant contributions to the field of fake news detection through various ML techniques and datasets, current research distinguishes itself through a comprehensive evaluation of multiple classifiers and feature selection methods using the diverse and extensive TruthSeeker dataset. This multifaceted approach provides a more detailed understanding of effective strategies for detecting fake news and advancing the state of the art in this critical area of research.

3. Methodology

This section highlights the stages followed in this investigation, including data collection, implementation of machine learning for the detection of fake news, selection of features and performance evaluation.

3.1. Data Collection

This study employs the TruthSeeker dataset, one of the largest ground-truth collections for fake news detection on Twitter, which spans from 2009 to 2022 and includes more than 180,000 labeled tweets. The dataset creation followed a structured multi-phase annotation process designed to ensure high reliability. Initial real and fake statements were collected from PolitiFact and keyword-based crawling was used to retrieve related tweets. The annotation process involved Amazon Mechanical Turk (MTurk), where each tweet-statement pair was evaluated by three independent Master Turkers. These highly qualified annotators completed semantic similarity tasks, judging how much a tweet agreed or disagreed with the source statement using predefined five-class and three-class labeling schemes. Only responses with a two-thirds majority agreement were retained to ensure label consistency.This robust process, grounded in active learning and human verification, makes TruthSeeker a reliable resource for both binary and multiclass fake news detection.

In addition to labels, the TruthSeeker dataset offers enriched information to enhance its analytical capabilities. Each tweet is supplemented with scores that assess bot likelihood, user credibility, and influence, providing a comprehensive view of the source and its potential impact. These scores are critical to understanding the mechanisms of automated misinformation, assessing content credibility, and studying the role of influential users in spreading false information. Furthermore, the dataset includes rich metadata, such as user profiles, tweet timestamps, and engagement metrics (likes, retweets, replies), offering invaluable insights into the spread and impact of fake news across networks.

The TruthSeeker dataset is made available through the Canadian Institute for Cybersecurity (CIC) as an open resource for researchers worldwide. Its high validation standards and extensive metadata make it indispensable for advancing fake news detection methodologies. Researchers can use it to develop and test new machine learning models, explore patterns in misinformation dissemination, and design interventions to mitigate its spread. In general, the TruthSeeker dataset is a meticulously validated and richly annotated resource that provides a solid foundation to tackle the complex issue of misinformation on social networks. Tables 1, 2, and 3 summarize the text, lexical, and metadata features of

Table 1: Text Features for Fake News Detection

Name	**Description**
Unique Count	Total count of unique, complex terms in text
Total Count	Overall word count in the text
ORG Percent	Percentage of text referencing organizations
NORP Percent	Percentage of text mentioning groups or affiliations
GPE Percent	Percentage of text related to geopolitical entities
PERSON Percent	Percentage of text including references to individuals
MONEY Percent	Percentage of text discussing monetary amounts
DATA Percent	Percentage of text containing references to dates
CARDINAL Percent	Percentage of text using cardinal numbers
PERCENT Percent	Percentage of text involving percentages
ORDINAL Percent	Percentage of text using ordinal numbers
LAW Percent	Percentage of text mentioning legal documents
PRODUCT Percent	Percentage of text referring to products
EVENT Percent	Percentage of text mentioning specific events
TIME Percent	Percentage of text referring to time-related details
LOC Percent	Percentage of text with references to locations
WORK OF ART Percent	Percentage of text mentioning artistic works
QUANTITY Percent	Percentage of text involving quantities
LANGUAGE Percent	Percentage of text referring to languages
Max Word	Length of the longest term in the text
Min Word	Length of the shortest term in the text
Avg Word Length	Average length of terms within the text

the dataset, respectively.

3.2. Feature Preprocessing: Standardization

An essential phase in the machine learning workflow is feature preprocessing, with *standardization* being one of the most significant techniques applied [26]. Standardization ensures that all features are rescaled to have a mean of 0 and a standard deviation of 1, which is vital for the effective functioning of many machine learning algorithms. This transformation is performed independently for each feature using the following formula:

$$x' = \frac{x - \mu}{\sigma} \tag{1}$$

Here, x represents the original feature value, μ is the mean of the feature, and σ denotes the standard deviation of the feature. After applying this formula, each feature is standardized to have a mean of 0 and a standard deviation of 1.

Standardization is particularly critical for algorithms that are sensitive to the scale of input data. Models such as Logistic Regression (LR), Support Vector Machines (SVM), and Neural Networks (NN) rely on consistent feature scaling to ensure effective learning. Without standardization, algorithms may disproportionately emphasize features with larger ranges, resulting in suboptimal model performance.

By normalizing the feature scales, these algorithms can focus on identifying the true relationships between the input features and the target variable, rather than being misled by differences in feature magnitudes. In addition, standardization improves the numerical stability and efficiency of optimization algorithms. Standardized features minimize the

Table 2: Lexical Features for Fake News Detection

Name	**Description**
Present Verb	Count of present tense verbs
Past Verb	Count of past tense verbs
Adjectives	Total number of adjectives in the text
Pronouns	Total number of pronouns in the text
TO's	Frequency of the word "to"
Determiners	Count of determiner words
Conjunctions	Total count of conjunctions
Dots	Instances of period (.) usage
Exclamations	Instances of exclamation marks (!) usage
Question	Instances of question marks (?) usage
Ampersand	Frequency of ampersand (&) usage
Capitals	Count of capitalized letters
Quotes	Total instances of quotation marks
Digits	Frequency of numerical digits (0-9)
Long Word Freq	Frequency of long words
Short Word Freq	Frequency of short words

Table 3: Meta-Data Features for Twitter Users

Name	**Description**
Followers Count	Total number of followers of the user
Friends Count	Total number of friends connected to the user
Favourites Count	Total likes given across all user tweets
Statuses Count	Overall count of tweets by the user
Listed Count	Number of times the user's tweets are listed
Mentions	Total mentions of the user in tweets
Quotes	Count of times the user's tweets were quoted
Replies	Count of replies received by the user
Retweets	Total retweets of the user's content
Favourites	Count of times the user's tweets were favorited
Hashtags	Number of hashtags used in the user's tweets
URLs	Whether the user's profile contains a URL
BotScoreBinary	Binary indicator of the user being a bot
Cred	Score measuring the user's credibility
Normalized Influence	Normalized score of the user's influence

risk of numerical errors, such as overflow or underflow, that can arise during complex calculations.

Another significant benefit of standardization is its role in accelerating optimization algorithms, such as gradient descent. When features are on a uniform scale, the optimization process becomes more stable and achieves convergence faster, helping the model to find the optimal solution efficiently.

3.3. Classification Techniques

This section outlines the classification techniques employed in the study, focusing on both individual classifiers and ensemble methods. The evaluation process aims to assess their effectiveness and identify the most suitable models for integration into ensemble frameworks. The classification approaches are categorized into two groups: base classifiers and ensemble methods, with the latter further divided into homogeneous and heteroge-

neous strategies.

3.3.1. Base Classifiers

The study begins by evaluating various individual classifiers to determine their effectiveness for inclusion in the ensemble. Identifying high-performing models is essential for constructing robust ensemble methods [27–29].

- *Decision Trees (DT)* use a tree-like structure to model decision-making processes, where nodes represent feature tests, branches denote outcomes, and leaves indicate class labels. The tree is built by partitioning the data based on attributes that maximize information gain. DTs are easy to interpret and visualize, accommodating both numerical and categorical features. However, they are susceptible to overfitting, especially with deep trees or small datasets [30, 31].

- *Naive Bayes (NB)* is a probabilistic classifier based on Bayes' theorem, often applied to text classification tasks. It assumes feature independence given the class, simplifies computations, and enables efficient training. Despite its simplicity, NB performs well in areas such as spam detection and sentiment analysis, but struggles with highly correlated features or non-Gaussian data distributions [32].

- *Support Vector Machines (SVMs)* are versatile algorithms that separate classes by finding the hyperplane that maximizes the margin between them. SVMs can handle linear and non-linear problems using kernel functions such as the linear, polynomial, and radial basis function (RBF). The regularization parameter C controls the trade-off between the margin size and the classification errors [33, 34].

- *K-Nearest Neighbors (KNN)* is a straightforward algorithm that assigns class labels based on the majority class among the K nearest neighbors. Distance metrics such as Euclidean or Manhattan are used to find neighbors. KNN is effective for multiclass classification, but can be computationally expensive for large datasets and is sensitive to values of K and the dimensionality of features [31, 35].

- *Random Forest (RF)* builds multiple decision trees on subsets of random data and aggregates their predictions. This ensemble approach reduces overfitting and variance, making RF suitable for noisy and high-dimensional data. It also provides feature importance scores, but can be computationally intensive for large datasets [36, 37].

- *Multilayer Perceptron (MLP)* is a type of neural network with multiple layers that capture nonlinear relationships in data. Using activation functions such as ReLU or sigmoid, MLPs can model complex patterns. However, they require careful regularization to avoid overfitting and can be computationally demanding [38].

3.3.2. Homogeneous Ensemble Classifiers

Homogeneous ensembles combine multiple models of the same type to improve predictive accuracy and reduce variance. These methods utilize the strengths of individual models to enhance stability and reliability. The following are key homogeneous ensemble techniques used in this study [28, 35].

- *Bagging (Bootstrap Aggregating)* creates diverse training sets by bootstrapping and trains multiple instances of a base model. The final prediction is determined by aggregating the output, typically using majority voting. Bagging is particularly effective in reducing overfitting in models prone to high variance, such as decision trees [27].
- *AdaBoost (Adaptive Boosting)* builds a sequence of models where each subsequent model focuses on correcting errors made by its predecessors. The final predictions are weighted on the basis of the models' performance. AdaBoost improves weak learners, but can be sensitive to noisy data and class imbalances [35].
- *Gradient Boosting* iteratively trains models to minimize residual errors from previous models. The final output combines all the models' predictions. This technique is highly flexible, supports various data types, and provides insight into the importance of features [27, 28].
- *XGBoost (Extreme Gradient Boosting)* improves traditional gradient boosting with optimizations such as parallel processing and regularization, making it computationally efficient and highly effective for complex datasets [39].
- *Extra Trees* generates an ensemble of extremely randomized decision trees. Introduces randomness in feature selection and split point determination, improving generalization and efficiency [39, 40].
- *CatBoost* specializes in handling categorical data through efficient encoding techniques. It includes mechanisms to prevent overfitting and supports GPU acceleration, making it suitable for diverse datasets.
- *Hist Gradient Boosting* employs histogram-based decision trees to improve computational efficiency by approximating continuous values with histograms. It is particularly useful for large-scale problems [27, 35].
- *Isolation Forests* focus on identifying anomalies by isolating outliers using random feature splits. Shorter isolation paths indicate potential anomalies, making this method valuable for detecting rare patterns in high-dimensional data [41, 42].

3.3.3. Heterogeneous Ensemble Classifiers

Heterogeneous ensembles combine diverse classifiers to leverage their complementary strengths, improving the accuracy and robustness of predictions. The key methods explored in this study include stacking and voting, both of which are effective in detecting fake news.

- *Voting Classifiers* aggregate predictions from different base models to create a unified output. Hard voting relies on majority class votes, while soft voting uses weighted probabilities based on model performance. Voting classifiers benefit from the diversity of base models, which enhances overall predictive performance [34].

- *Stacking Classifiers* employ a meta-learner to combine predictions from multiple base models. Unlike voting, stacking uses a data-driven approach to learn the optimal combination of models output. Using the strengths of various models, stacking often achieves superior performance compared to individual models or simple voting schemes [34, 35].

3.4. Feature Selection Methods

Feature selection is a crucial step in machine learning that aims to identify the most relevant features for model building. Effective feature selection not only improves model performance but also reduces computational cost and helps prevent overfitting. In this context, several techniques can be employed to select the most informative features. Three prominent methods are Recursive Feature Elimination (RFE), SelectKBest, and Principal Component Analysis (PCA). Each of these methods utilizes distinct principles and mathematical foundations to achieve feature selection.

3.4.1. Recursive Feature Elimination

RFE is an iterative feature selection technique that recursively removes the least significant features based on the performance of a chosen model. The algorithm continues to remove features until only the desired number of features remains. RFE works by fitting a model to the data and ranking the features based on their importance. The features are then removed on the basis of this ranking, and the model is re-trained on the reduced feature set. This process is repeated until the number of features is equal to the target number, k. RFE is typically used with models that provide feature importance scores, such as LR, SVM, or DTs.

- **Steps of RFE:**

  (i) **Model Training:** Train a model M on the initial set of features X to predict the target y.

  (ii) **Feature Ranking:** Evaluate the importance of each feature based on the coefficients of the model or feature weights.

  (iii) **Feature Elimination:** Remove the least important feature(s) from the feature set.

  (iv) **Model Retraining:** Re-train the model with the reduced set of features.

  (v) **Iteration:** Repeat the process until the desired number of features k is reached.

- **Mathematical Formulation:**

  Let X be the matrix of features with samples n and features m, and y be the target vector.

  (i) Model Construction: Train a model M with features X:

  $$f(X) = \hat{y} \tag{2}$$

  where $\hat{y}$ is the predicted target vector.

  (ii) Feature Importance: Compute the importance of each feature i:

  $$\text{Feature Importance} = |w_i| \tag{3}$$

  where w_i is the coefficient for feature i.

  (iii) Feature Elimination: Identify and remove the least important feature(s) based on $|w_i|$.

  (iv) Update Feature Set: Update the feature set X and repeat the training and elimination steps.

3.4.2. SelectKBest

SelectKBest is a univariate feature selection method that selects the top k features based on univariate statistical tests. It evaluates each feature independently to determine its relevance to the target variable. SelectKBest evaluates each feature using statistical tests and selects the top k features based on test scores. Common statistical tests used include the chi-square test, the ANOVA F-value, and mutual information.

- **Steps of SelectKBest:**

  (i) **Compute Test Statistics:** Calculate the test statistic for each feature based on a chosen statistical test.

  (ii) **Rank Features:** Rank features according to their test statistics.

  (iii) **Select Top Features:** Choose the best k features with the highest test statistics.

- **Mathematical Formulation:**

  Let X be the matrix of features with samples n and features m, and y be the target vector.

  (i) Compute Test Statistics: For each feature X_i, compute the test statistic T_i:

  $$T_i = \text{Test Statistic}(X_i, y) \tag{4}$$

  (ii) Ranking Features: Rank features based on the computed test statistics:

  $$\text{Rank}(X_i) = \text{Sort}(T_1, T_2, \ldots, T_m) \tag{5}$$

  (iii) Selecting Top k: Select the k features with the highest T_i:

  $$X_{\text{selected}} = \text{Top } k \text{ features based on } T_i \tag{6}$$

3.4.3. Principal Component Analysis

PCA is a dimensionality reduction technique that transforms the original feature space into a new set of uncorrelated variables called principal components. PCA identifies the directions (principal components) along which the variance of the data is maximized. PCA transforms data into a new coordinate system where the first coordinate accounts for the largest variance in the data, the second coordinate accounts for the next largest variance, and so on. The goal is to reduce the number of dimensions while retaining as much variance as possible.

- **Steps of PCA:**
  - (i) **Compute Covariance Matrix:** Calculate the covariance matrix of the feature set.
  - (ii) **Eigen Decomposition:** Perform eigenvalue decomposition to obtain eigenvectors and eigenvalues.
  - (iii) **Sort Principal Components:** Sort eigenvectors by the magnitude of their corresponding eigenvalues.
  - (iv) **Select Top Components:** Choose the top k eigenvectors to form the principal components.
  - (v) **Project Data:** Transform the original data into the new feature space spanned by the selected principal components.
- **Mathematical Formulation:**

  Let X be the feature matrix with n samples and m features.

  - (i) Compute Covariance Matrix:

$$\Sigma = \frac{1}{n-1} X^T X \tag{7}$$

  - (ii) Perform Eigen Decomposition: Find eigenvectors W and eigenvalues Λ from the covariance matrix:

$$\Sigma W = W\Lambda \tag{8}$$

  - (iii) Sort Eigenvectors: Sort eigenvectors based on the eigenvalues in descending order.
  - (iv) Select the top k components: Choose the top k eigenvectors to form the new feature space.
  - (v) Project Data: Project the original data onto the new space:

$$X_{\text{new}} = XW \tag{9}$$

3.4.4. Genetic Algorithm

GAs simulate the process of natural evolution to solve optimization problems. They operate on a population of potential solutions and use evolutionary processes such as selection, crossover, and mutation to evolve better solutions over generations.

- **Mathematical Formulation**

  The GA-based feature selection process involves the following components:

- **Representation**

  Each individual in the GA population represents a subset of features using a binary vector. If the i-th bit of the vector is 1, the corresponding feature is included in the subset; otherwise, it is excluded. Let X be the feature matrix with n features and y the target variable. Each individual $\mathbf{x}$ can be represented as:

$$\mathbf{x} = [x_1, x_2, \ldots, x_n] \tag{10}$$

  where $x_i \in \{0, 1\}$.

- **Fitness Function**

  The fitness function evaluates the quality of the subset of features. In our case, the fitness function is the accuracy of a classifier (Random Forest in this work) trained on the selected features:

$$f(\mathbf{x}) = \text{Accuracy}(\text{Classifier}(X_{\mathbf{x}}, y)) \tag{11}$$

  where $X_{\mathbf{x}}$ denotes the feature matrix with features selected according to $\mathbf{x}$.

- **Genetic Algorithm Operations**

  The GA operations are as follows:

  (i) **Initialization**: Generate an initial population of binary vectors.

  (ii) **Selection**: Select individuals based on their fitness using tournament selection.

  (iii) **Crossover**: Create new individuals by exchanging segments of binary vectors between parents.

  (iv) **Mutation**: Introduce variability by flipping bits in the binary vectors.

  (v) **Replacement**: Form the next generation of the population.

3.5. Performance Measures

In assessing the performance of the previously outlined classification ML methods for the detection of fake news in tweets, it is crucial to evaluate their accuracy, recall, precision, F1 score and AUC-ROC. These metrics provide valuable insights into the model's ability to correctly classify tweets as either real or fake, detect relevant instances, balance between precision and recall, and evaluate the overall performance of the classification model. Given the potential impact of false positives - where legitimate news can be incorrectly labeled fake - it is particularly important to evaluate models not only based on overall accuracy, but also in terms of their precision and recall. High precision helps minimize false positives, which is crucial for maintaining trust and preventing the suppression of valid content. In contrast, high recall ensures that actual instances of fake news are effectively identified. The combination of these metrics supports a balanced and context-aware interpretation of the behavior of the model, allowing stakeholders to prioritize detection objectives based on specific deployment needs.

- *Accuracy* is a measure of how well a model is able to correctly classify tweets as real or fake. It is calculated as the proportion of correctly classified instances (both real and fake tweets) out of the total number of instances.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{12}$$

  where:

  - TP (True Positives) refers to the number of fake tweets correctly classified as fake.
  - TN (True Negatives) refers to the number of real tweets correctly classified as real.
  - FP (False Positives) refers to the number of real tweets incorrectly classified as fake.
  - FN (False Negatives) refers to the number of fake tweets incorrectly classified as real.

- *Precision* measures the model's ability to avoid false positives, focusing on how many of the tweets classified as fake are indeed fake. It is calculated as the proportion of true positives out of the total number of instances classified as fake.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{13}$$

  In the context of fake news detection in tweets, precision indicates the model's effectiveness in identifying actual fake tweets among those it classifies as fake.

- *Recall* measures the proportion of actual fake tweets that are correctly identified by the model. It is calculated by dividing the number of true positives by the sum of true positives and false negatives.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{14}$$

  In the context of fake news detection in tweets, recall indicates the model's ability to detect all the fake tweets that are actually present.

- *F1 Score* is a harmonic mean of precision and recall, providing a balanced measure of the model's performance in detecting fake tweets. It accounts for both the precision of the model's predictions and its ability to capture all relevant instances.

$$\text{F1} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{15}$$

  The F1 score is particularly useful in the detection of fake news for tweets where both false positives and false negatives have significant consequences, and the balance between precision and recall is crucial.

- *AUC-ROC (Area Under the Receiver Operating Characteristic Curve)* is a performance measurement for classification problems at various thresholds. It plots the true positive rate (recall) against the false positive rate and measures the area under this curve.

$$\text{AUC-ROC} = \int_{-\infty}^{+\infty} \text{TPR}(x)\, \text{dFPR}(x) \tag{16}$$

  where:

  – TPR (True Positive Rate) is the recall of the model.
  – FPR (False Positive Rate) is the proportion of actual negatives that are incorrectly classified as positives.

  In the context of fake news detection in tweets, the AUC-ROC score provides an aggregate measure of the model's ability to discriminate between fake and real tweets across all possible classification thresholds. A higher AUC-ROC score indicates a better overall performance in distinguishing between fake and real tweets.

4. Results and Analysis

4.1. Individual Classifiers Results

Table 4 demonstrates the performance of various individual ML models in detecting fake news using the TruthSeeker dataset. The results shown in this table reveal notable

insights into the effectiveness of these models in detecting fake news. Each model was assessed based on five key metrics: accuracy, precision, recall, F1 score, and AUC-ROC, which together provide a comprehensive view of their predictive capabilities. In this regard:

Table 4: Performance of Individual Classifiers on the TruthSeeker Dataset

Classifier	Accuracy	Precision	Recall	F1-score	AUC-ROC
NB	92.969%	92.996%	92.969%	92.971%	93.414%
LR	95.801%	95.801%	95.801%	95.801%	95.889%
KNN	92.414%	92.414%	92.414%	92.414%	95.029%
MLP	95.387%	95.387%	95.387%	95.387%	95.847%
SVM	95.704%	95.704%	95.704%	95.704%	95.902%
DT	91.006%	91.007%	91.006%	91.006%	91.002%
RF	95.801%	95.801%	95.801%	95.801%	95.892%

- **Naive Bayes:** NB achieved an accuracy of 92.969%, with precision and recall close to 93%, indicating its ability to consistently identify both true positives and true negatives. Its AUC-ROC of 93.414% reflects a robust performance in distinguishing between fake and real news.

- **Logistic Regression:** LR demonstrated superior performance with accuracy, precision, recall, and F1 score of 95.801%. The AUC-ROC score of 95.889% further emphasizes its strong discriminatory power, making it one of the most reliable classifiers in this evaluation.

- **K-Nearest Neighbors:** KNN showed a slightly lower performance with an accuracy of 92.414%. Despite its consistent scores in precision, recall, and F1 score, its AUC-ROC of 95.029% suggests that KNN can effectively handle the classification task, although not as efficiently as some other methods.

- **Multilayer Perceptron:** MLP achieved an accuracy of 95.387%, with matching precision, recall, and F1 score values. Its AUC-ROC of 95.847% indicates strong predictive performance, making it a competitive choice for the detection of fake news.

- **Support Vector Machine:** SVM stood out with an accuracy of 95.704% and uniformly high scores for precision, recall, and F1 score. Its AUC-ROC of 95.902% underscores its excellent ability to differentiate between classes, proving to be one of the best performers.

- **Decision Tree:** DT showed a lower overall performance with an accuracy of 91.006%. Its precision and recall values were slightly higher at 91.007%, but the F1 score and AUC-ROC of 91.002% reflect its limitations in handling the data set as effectively as other classifiers.

- **Random Forest:** RF matched LR in accuracy, precision, recall, and F1 score at 95. 801%. Its AUC-ROC of 95.892% confirms its reliability and robustness, showcasing its potential as a powerful ensemble method for detecting fake news.

In summary, LR, SVM and RF emerged as the leading classifiers, demonstrating high precision and balanced performance across all metrics. These methods provide a solid foundation for reliable fake news detection on the TruthSeeker dataset, highlighting their practical applicability in real-world scenarios.

4.2. Ensemble Methods Results

Table 5 presents the performance metrics of various ensemble ML algorithms used for the detection of fake news. These algorithms include both homogeneous and heterogeneous ensemble methods, evaluated across five key metrics: Accuracy, precision, recall, F1 score, and AUC-ROC. The results depicted in this table can be summarized as follows:

- **CatBoost** demonstrated the highest overall performance with an accuracy of 95.801% and an equally high precision, recall, and F1 score of 95.801%. Its AUC-ROC score of 95.859% highlights it as one of the main methods to distinguish between fake and real news.
- **HistGradient** Boosting also achieved a high accuracy of 95.801% with consistent metrics for precision, recall, and F1 score. Its AUC-ROC of 95.904% is the highest among all methods tested, marking it as the most effective classifier for this dataset.
- **LightGBM** matched CatBoost and HistGradient Boosting with an accuracy of 95.801% and consistent Precision, Recall and F1 score. The AUC-ROC score of 95.854% further establishes it as a highly effective method for the detection of fake news.
- **XGBoost** achieved an accuracy of 95.790% with an equivalent precision, recall, and F1 score. Its AUC-ROC of 95.880% reflects strong performance and makes it a competitive choice among the tested classifiers.
- **Stacking Classifier** produced an accuracy of 95.801% and matched the metrics for Precision, Recall, and F1-score. With an AUC-ROC of 95.816%, it is also a strong performer for the detection of fake news.
- **Voting Classifier** showed an accuracy of 95.786% with consistent Precision, Recall, and F1-score values. The AUC-ROC of 95.892% indicates reliable performance, although slightly less effective compared to the best methods.
- **Bagging (Decision Tree)** demonstrated high performance with an accuracy of 95.738% and uniform scores for precision, recall, and F1 score. Its AUC-ROC of 95.893% confirms its effectiveness in handling the classification task.
- **Gradient Boosting** provided strong results with an accuracy of 95.801% and consistent metrics for precision, recall, and F1 score. Its AUC-ROC of 95.827% reflects effective performance for fake news detection.

- **Extra Trees** achieved an accuracy of 95.756% with very close values for precision, recall, and F1 score. The AUC-ROC of 95.779% indicates strong performance, although slightly lower than the top methods.
- **AdaBoost (Decision Tree)** showed a lower performance with an accuracy of 91.025% and consistent scores on all metrics. The AUC-ROC of 91.019% indicates that it is less effective compared to other methods.
- **Isolation Forest** was the least effective with an accuracy of 51.140% and significantly lower scores for precision, recall, and F1 score. The AUC-ROC of 47.959% reflects its unsuitability for the detection of fake news, as it is designed for the detection of anomalies rather than classification.

Table 5: Performance of Ensemble Classifiers on the TruthSeeker Dataset

Classifier	**Accuracy**	**Precision**	**Recall**	**F1-score**	**AUC-ROC**
AdaBoost (Decision Tree)	91.025%	91.025%	91.025%	91.025%	91.019%
Bagging (Decision Tree)	95.738%	95.738%	95.738%	95.738%	95.893%
CatBoost	95.801%	95.801%	95.801%	95.801%	95.859%
Extra Trees	95.756%	95.757%	95.756%	95.756%	95.779%
Gradient Boosting	95.801%	95.801%	95.801%	95.801%	95.827%
HistGradient Boosting	95.801%	95.801%	95.801%	95.801%	95.904%
Isolation Forest	51.140%	26.376%	51.140%	34.802%	47.959%
LightGBM	95.801%	95.801%	95.801%	95.801%	95.854%
Stacking Classifier	95.801%	95.801%	95.801%	95.801%	95.816%
Voting Classifier	95.786%	95.786%	95.786%	95.786%	95.892%
XGBoost	95.790%	95.790%	95.790%	95.790%	95.880%

In summary, the best-performing methods for fake news detection are CatBoost, HistGradient Boosting, LightGBM, and XGBoost. Among these, HistGradient Boosting stands out with the highest AUC-ROC score, while CatBoost and LightGBM also demonstrate exceptional performance across all metrics. The Stacking Classifier and Voting Classifier are also effective but slightly less so compared to the aforementioned top methods. Bagging (Decision Tree) and Gradient Boosting offer strong performance, though they fall short of the best methods. Extra Trees and AdaBoost (Decision Tree) show solid results but are not as competitive, while Isolation Forest proves to be inadequate for this specific classification task.

Given that individual classifiers and ensemble methods demonstrated comparable levels of performance and considering the reduced computational complexity associated with individual models, the subsequent analysis focuses on evaluating the impact of feature selection techniques on individual classifiers. In particular, feature selection methods such as RFE and GA were applied exclusively to individual classifiers in this study, and not to ensemble models. This methodological choice was made to isolate and assess the contribution of feature selection strategies in improving model performance while maintaining computational efficiency. The resulting analysis offers valuable information on the potential of optimized individual classifiers to deliver competitive performance with reduced

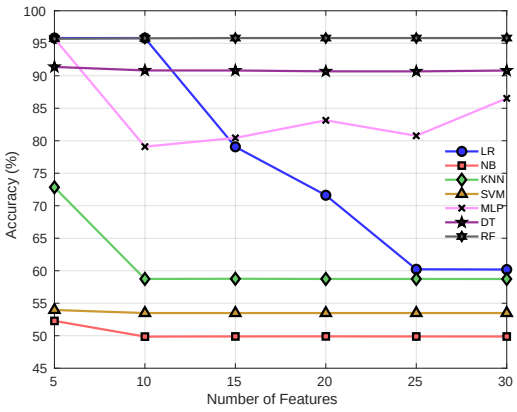
training overhead, thus supporting their applicability in resource-constrained or real-time environments.

4.3. Impact of Feature Selection on Machine Learning Algorithms

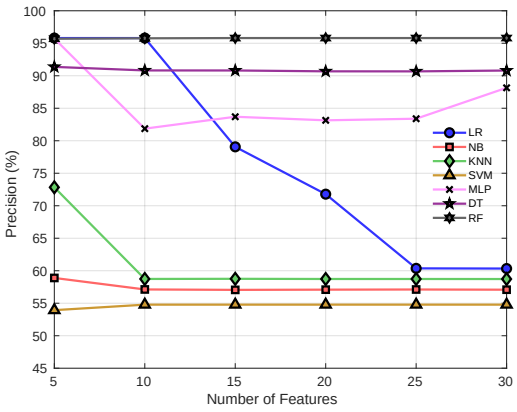
Feature selection plays a crucial role in enhancing the performance and efficiency of ML models. This section explores the impact of different feature selection techniques on the performance of various classifiers in the context of fake news detection. Specifically, the section investigates how the performance of ML algorithms is affected by selecting different subsets of features and reducing dimensionality. The analysis begins by evaluating the effectiveness of feature selection methods such as RFE and SelectKBest. These methods are examined by varying the number of features from 5 to 30 and assessing the impact on the performance metrics of classifiers including LR, NBs. Performance metrics such as accuracy, precision, recall, F1 score, and AUC-ROC are used to gauge how the increasing number of features influences the effectiveness of these models. Subsequently, PCA is explored as a technique for dimensionality reduction. The performance of classifiers is analyzed as the number of principal components is varied from 5 to 30, providing insights into how dimensionality reduction affects model performance. Finally, the section covers the use of GA for feature selection, focusing on how the optimal feature subset identified by the GA affects the performance of ML algorithms. The GA approach is evaluated on the basis of the optimal number of features selected and compared with the results obtained from RFE, SelectKBest, and PCA. Through this examination, the purpose of this section is to provide a comprehensive understanding of how different feature selection strategies and feature dimensions impact the performance of ML algorithms.

4.3.1. Recursive Feature Elimination Results

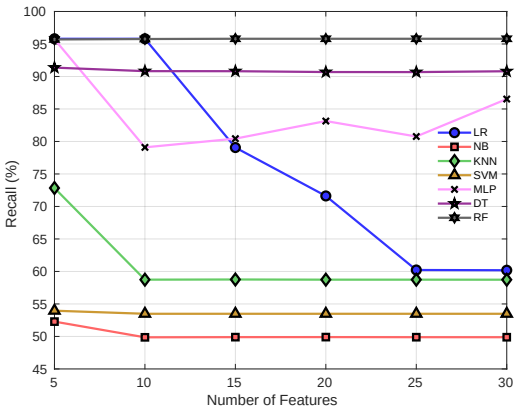
Figure 1 shows the performance of various machine algorithms as the number of selected features increases from 5 to 30 using the RFE feature selection method. For example, Figure 1a illustrates the accuracy of the model as the number of features increases from 5 to 30. The accuracy results for different models using RFE exhibit various trends as the number of features increases. LR maintains a high accuracy of 95.80% with 5 features, which remains constant up to 10 features, but then decreases to 60.19% with 30 features. NB shows a low and relatively constant accuracy in all feature sets, with 52.28% at 5 features and 49.88% at 30 features. KNN starts with a higher accuracy of 72.84% at five features and then stabilizes at 58.73% from 10 features onward. SVM shows a consistently low accuracy of around 53.49% in all feature sets, indicating a minimal sensitivity to the number of features. MLP starts with a high accuracy of 95.73% at 5 features, decreases to 79.09% at 10 features, and fluctuates before increasing to 86.53% at 30 features. DT exhibits high and stable accuracy, ranging from 91.36% at 5 features to 90.80% at 30 features. Finally, RF maintains a high and consistent accuracy from 95.70% at 5 features to 95.80% at 30 features, demonstrating the most stable performance. On the other hand, figure 1b shows precision values as the number of features increases.



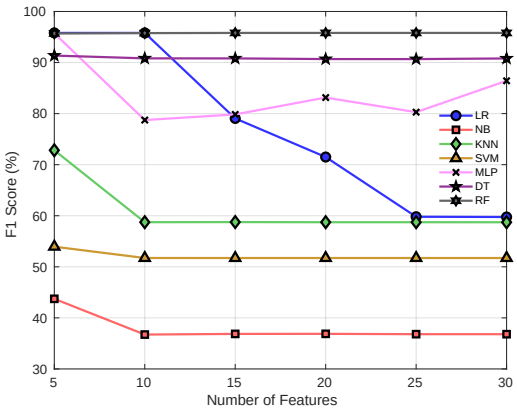
a Accuracy



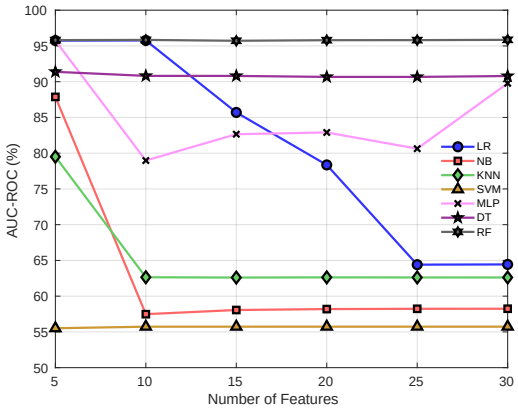
b Precision



c Recall



d F1 Score



e AUC-ROC

Figure 1: Performance Metrics of Various Models as Feature Number Increases under Recursive Feature Elimination.

The precision results for different models using RFE show a varied performance in terms of number of features. For LR, the precision starts at 95.80% with 5 features and decreases to 60.34% as the number of features increases to 30. NB shows a relatively stable precision, with 58.89% at 5 features and 57.08% at 30 features. KNN maintains a precision level from 72.83% at 5 features to 58.72% at 30 features, indicating a gradual decrease. The SVM has stable but low precision, starting at 53.93% and slightly increasing to 54.78% as more features are used.

MLP begins with a high precision of 95.73% at 5 features, which drops to 81.86% at 10 features, and then fluctuates before increasing to 88.14% at 30 features. DT shows high and consistent precision throughout, with values from 91.37% to 90.80% throughout the range of features. RF exhibits high and steady precision across all set of features, starting at 95.70% and remaining at 95.80%.

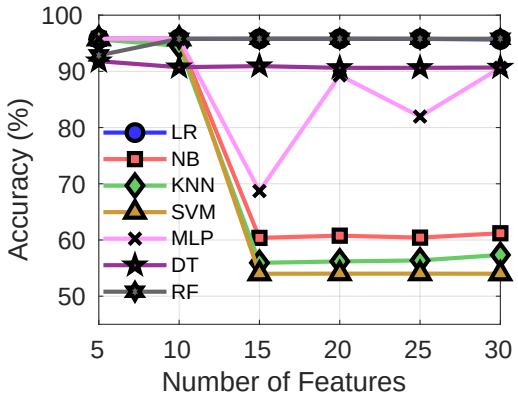
The recall results for different models using RFE display diverse trends as the number of features increases as shown in figure 1c. LR achieves a high recall of 95.80% in both 5 and 10 features, but experiences a significant decline to 60. 19% in 30 features. NB shows a consistently low recall, with values of 52.28% at five features and remaining around 49.88% at 30 characteristics. KNN begins with a high recall of 72.84% for five features and stabilizes at 58.73% for feature counts of 10 and above. SVM maintains a constant recall of 53.49% across all feature sets, indicating that it is relatively unaffected by the number of features. MLP starts with a high recall of 95.73% at 5 features, decreases to 79.09% at 10 features, and then fluctuates before rising to 86.53% at 30 features, showing an overall trend of improvement with more features. DT shows a high and stable recall from 91.36% at 5 features to 90.80% at 30 features, demonstrating robust performance regardless of the number of features. Lastly, RF also shows high and consistent recall from 95.70% with 5 features to 95.80% with 30 features, representing the most reliable and effective model in terms of recall. The results of the F1 score for various models using RFE reveal different trends as the number of features increases as depicted in figure 1d. LR maintains a high F1 score of 95.80% for 5 and 10 features, but shows a notable decrease to 59.75% for 30 features, indicating a decrease in performance as more features are added. NB exhibits low F1 scores throughout, starting at 43.73% with 5 features and stabilizing around 36.80% from 10 to 30 features, reflecting poor performance with increasing features. KNN achieves a high F1 score of 72.82% for five features, but then shows a decrease to about 58.73% for feature counts of 10 and above. SVM demonstrates a relatively consistent F1 score of 51.72% in most feature sets, with a slight peak at 53.94% with 5 features. MLP starts with a high F1 score of 95.73% at 5 features, drops to 78.74% at 10 features, and fluctuates before reaching 86.43% at 30 features, suggesting improved performance with more features. DT shows high and stable F1 scores from 91.36% at 5 features to 90.80% at 30 features. Lastly, RF maintains high and consistent F1 scores from 95.70% at 5 features to 95.80% at 30 features, reflecting the most effective and reliable performance in terms of the F1 score.

The AUC-ROC results for various models using RFE illustrate different performance trends as the number of features increases, as illustrated in Figure 1e. LR and MLP both achieve high AUC-ROC scores of 95.75% at 5 features. However, while LR exhibits

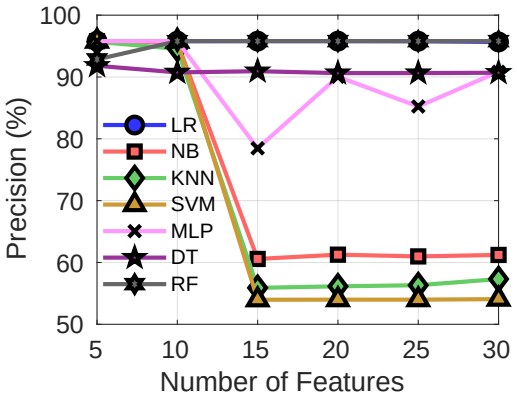
a decrease in AUC-ROC to 64.44% by 30 features, MLP maintains a high AUC-ROC with scores of 89.78% at 10 features and 89.78% at 30 features. NB shows a significant drop in AUC-ROC from 87.87% at 5 features to 58.24% at 30 features. KNN achieves a maximum AUC-ROC of 79.51% at 5 features but stabilizes around 62.61% from 10 features onward. SVM consistently shows a low AUC-ROC score of 55.73% in all feature sets. DT demonstrates stable AUC-ROC scores of approximately 90.80% on all feature counts. Finally, RF maintains the highest and most consistent AUC-ROC scores, ranging from 95.81% at 5 features to 95.85% at 30 features.

4.3.2. SelectKBest Restlts

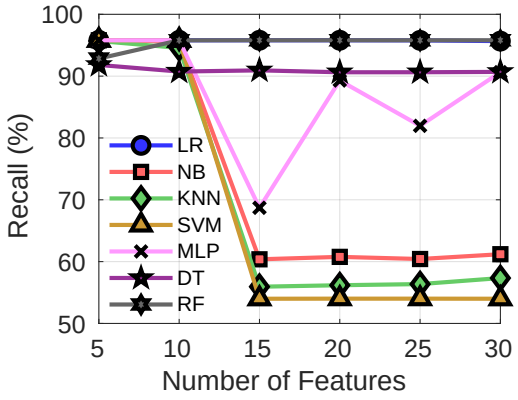
Figure 2 illustrates the impact of feature selection on the performance of various ML algorithms as the number of selected features increases from 5 to 30. As shown in the figure 2a, the accuracy of various ML models is affected by increasing the number of features selected using the SelectKBest method from 5 to 30. For LR, DT, and RF, the accuracy remains relatively stable across the range of features. Specifically, LR maintains a high accuracy of approximately 95.80% throughout the feature selection process. SVM exhibits a significant decrease in accuracy, dropping from 95.80% at 5 features to 53.99% at 30 features, indicating a considerable decline in performance as the number of features increases. In contrast, NB and KNN also show a decrease in accuracy, but not as drastic as SVM, with NB falling from 95.80% at 5 features to around 61.20% at 30 features and KNN decreasing from 95.69% at 5 features to 57.34% at 30 features. MLP demonstrates variable performance, with a notable drop in 15 features but improved to around 90.64% at 30 features. Overall, most models achieve optimal or near-optimal accuracy with a higher number of features, except for SVM, which experiences a significant drop in performance. Figure 2b illustrates how the precision of various ML models changes as the number of features selected using the SelectKBest method increases from 5 to 30. For LR, DT, and RF, the precision remains relatively high and stable throughout the range of feature selection. LR maintains a consistent precision of approximately 95.80%, and RF also shows high precision around 95.80% throughout the feature selection process. In contrast, NB exhibits a significant decline in precision from 95.80% at 5 features to approximately 61.22% at 30 features, with only minor fluctuations at intermediate feature counts. KNN also experiences a decrease in precision from 95.69% with 5 features to 57.32% with 30 features. The MLP shows a fluctuating precision, dropping to 78.46% at 15 features but recovering to around 90.84% at 30 features. The DT maintains relatively constant precision around 90.63% to 90.69% throughout the feature range. Overall, most models see stable or improved precision as the number of features increases, with notable exceptions for NB and KNN, which exhibit considerable variability.



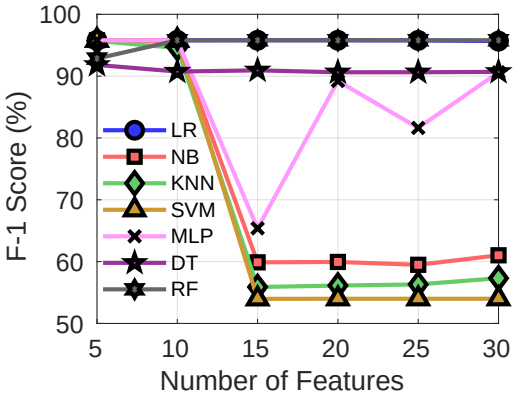
a Accuracy



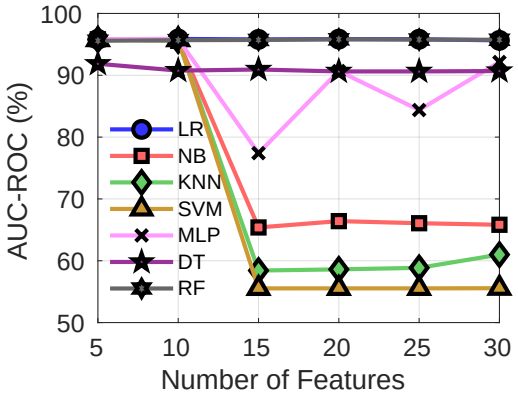
b Precision



c Recall



d F1 Score



e AUC-ROC

Figure 2: Performance Metrics of Various Models as Feature Number Increases under SelectKBest.

Figure 2c illustrates how the recall of various ML models is affected by increasing the number of features selected using the SelectKBest method from 5 to 30. For LR, MLP, DT, and RF, recall remains relatively high and stable throughout the range of features. LR and RF maintain high recall values, remaining above 95.66% for all feature counts. DT also shows stable high recall around 90.63% to 90.69%. In contrast, NB and KNN exhibit significant variability in recall. NB starts with a high recall of 95.80% at 5 features but drops to 61.20% at 30 features, with a slight increase at 30 features compared to 15 and 25 features. KNN also shows a decrease in recall from 95.69% at 5 features to 57.34% at 30 features, reflecting a steady decline in performance. SVM shows a generally low recall across all feature counts, starting at 95.80% at 5 features and decreasing to 53.99% at 30 features. MLP exhibits a notable increase in recall from 68.71% at 15 features to 90.64% at 30 features. In general, most models maintain or improve recall with more features, except for NB, KNN, and SVM, which show significant declines in recall performance.

Figure 2d shows the F-1 scores of various ML models as the number of features selected using the SelectKBest method increases from 5 to 30. LR maintains a consistently high F-1 score close to 95.80% under all feature counts. RF also shows a stable high F-1 score around 95.80%, demonstrating consistent performance regardless of the number of features. NB and KNN exhibit notable decreases in F-1 score as the number of features increases. NB starts with a high F-1 score of 95.80% at five features but decreases to around 61.02% at 30 features. Similarly, KNN shows a drop in the F-1 score from 95.69% at 5 features to 57.32% at 30 features, reflecting a steady decline in performance. SVM maintains a low and relatively stable F-1 score throughout the feature range, starting at 95.80% at 5 features and dropping to 54.00% at 30 features, showing a significant decline in performance as the number of features increases. MLP exhibits fluctuations in the F-1 score, with a significant drop to 65.38% at 15 features but recovered to approximately 90.62% at 30 features. DT maintains a relatively stable F-1 score of around 90.63% to 90.69% throughout the feature selection process. In general, most models, except NB, KNN and SVM, maintain or improve their F-1 scores as the number of features increases. NB and KNN show significant decreases, while SVM's performance declines substantially.

Figure 2e presents the AUC-ROC scores for different ML models as the number of features selected using the SelectKBest method increases from 5 to 30. LR maintains a high and relatively stable AUC-ROC score, starting at 95.83% with 5 features and slightly decreasing to 95.65% with 30 features. Similarly, RF exhibits a high and stable AUC-ROC score, remaining between 95.58% and 95.74% in the range of features. NB and KNN demonstrate a consistent decline in AUC-ROC as more features are added. NB has an initial AUC-ROC of 95.82% at 5 features, but this score drops to 65.82% at 30 features. KNN also shows a decrease from 95.77% at 5 features to 60.99% at 30 features, reflecting a clear downward trend in performance. SVM maintains a lower and more stable AUC-ROC score across all feature counts, starting at 95.78% at 5 features and decreasing marginally to 55.56% at 30 features. This indicates a steady decline in SVM's performance as more features are considered. MLP shows significant variability in AUC-ROC, with a marked drop to 77.41% at 15 features but recovered to 92.15% at 30 features, showing that the performance of this model is sensitive to the number of features selected. DT maintains

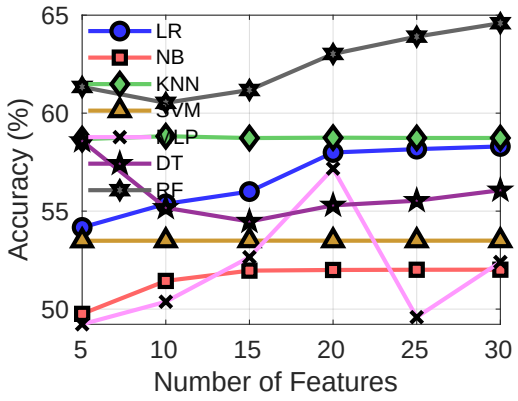
a relatively stable AUC-ROC score between 90.62% and 90.93% throughout the feature selection process, indicating consistent performance with the increasing number of features. Most models other than NB and KNN exhibit stable or improved AUC-ROC scores as the number of features increases. The performance of NB and KNN deteriorates significantly, while SVM shows a consistent decline, and MLP exhibits fluctuating performance.

Evaluation of ML models using four different performance metrics: accuracy, precision, recall, and AUC-ROC reveals distinct patterns in how these models perform as the number of features increases from 5 to 30 using the SelectKBest method. LR and RF exhibit robust and consistent performance across all metrics. For accuracy and AUC-ROC, LR remains stable at approximately 95.80% and 95.83% respectively, while RF also maintains high performance with accuracy around 95.80% and AUC-ROC between 95.58% and 95.74%. MLP shows variability in performance, with fluctuating precision, recall, and AUC-ROC, reflecting the sensitivity to the number of features selected.

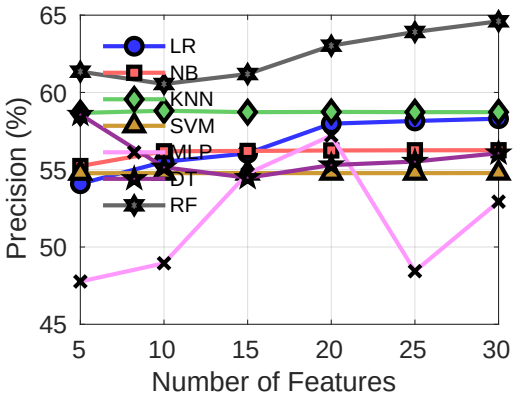
In contrast, NB and KNN show significant declines in performance metrics as the number of features increases. NB experiences substantial drops in precision, recall, and AUC-ROC from high initial values to around 61.02%, 65.82%, and 65.82% respectively, at 30 features. Similarly, KNN demonstrates a decrease in accuracy from 95.69% to 57.32%, precision from 95.69% to 57.32%, and AUC-ROC from 95.77% to 60.99% over the same range. SVM consistently shows lower performance across all metrics, with a substantial decrease in recall and AUC-ROC as the number of features increases, and a lower AUC-ROC value from 95.78% to 55.56%. Finally, DT exhibits stable performance in accuracy, precision and recall, maintaining a consistent AUC-ROC around 90.62% to 90.93%.

4.3.3. Principal Component Analysis Results

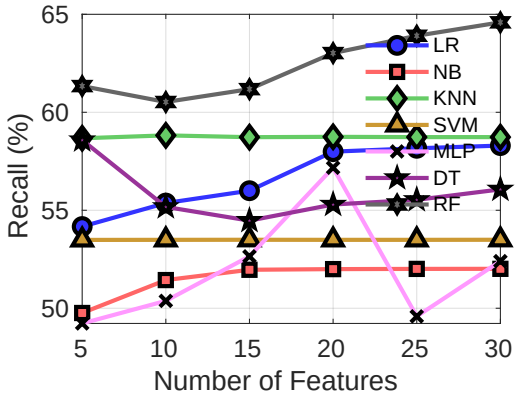
This section shows the results of applying PCA to reduce dataset dimensionality and its effects on the performance of various ML algorithms. Examines how increasing the number of principal components from 5 to 30 influences the accuracy, precision, recall, and AUC-ROC of the models. Figure 3a illustrates the accuracy of various ML models as the number of principal components increases from 5 to 30. The accuracy results obtained through PCA are generally lower compared to those of other feature selection methods. RF achieves the highest accuracy, with an increase from 61.34% at 5 components to 64.59% at 30 components, demonstrating the best performance among the models evaluated. In contrast, MLP exhibits the lowest accuracy, starting at 49.23% at 5 components and slightly improving to 52.40% at 30 components. KNN and SVM maintain relatively stable accuracies around 58.67% and 53.50%, respectively, while LR shows a gradual increase from 54.17% to 58.30%. DT demonstrates relatively lower performance, with accuracy values ranging from 58.58% to 56.07%. Overall, the use of PCA for dimensionality reduction results in lower accuracy compared to other feature selection methods, indicating that while PCA may reduce dimensionality, it does not necessarily enhance model performance as effectively as methods like RFE or SelectKBest.



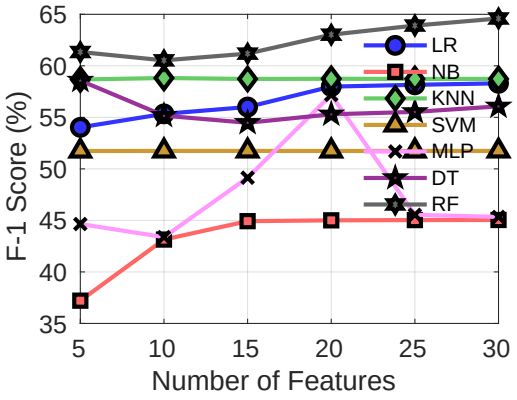
a Accuracy



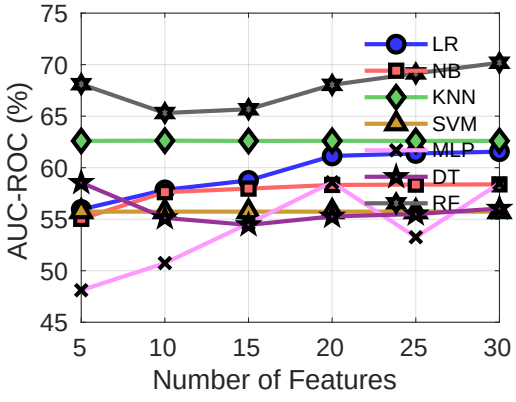
b Precision



c Recall



d F1 Score



e AUC-ROC

Figure 3: Performance Metrics of Various Models under PCA.

Figure 3b illustrates the precision of various ML models as the number of principal components increases from 5 to 30. The precision results for PCA are generally low compared to other feature selection methods, highlighting a notable deficiency in performance for most models. RF achieves the highest precision, increasing from 61.34% at 5 components to 64.60% at 30 components, indicating that it is the most effective model in terms of precision under PCA. In contrast, MLP exhibits the lowest precision, with a substantial drop from 47.77% at 5 components to 52.93% at 30 components, which shows that this model performs poorly in terms of precision. KNN and SVM maintain relatively stable precision values around 54.78% and 58.73%, respectively. LR and NB show a general increase in precision as the number of components grows, though their performance remains lower than that of RF. Specifically, LR improves from 54.09% to 58.30%, and NB shows a slight increase from 55.24% to 56.26%. Overall, the precision results underscore that PCA's dimensionality reduction approach leads to relatively poor performance across the models, with RF being the exception and MLP showing significant performance issues. Figure 3c shows the recall values for different MLP models as the number of principal components increases from 5 to 30. The recall values, which represent the proportion of true positive instances identified by each model, reveal a generally low performance across the models compared to other feature selection methods. RF exhibits the highest recall, increasing from 61.34% at 5 components to 64.59% at 30 components. This indicates that RF performs the best in identifying true positive instances as the number of principal components grows. KNN shows stable recall values, remaining around 58.67% to 58.73% throughout the range of principal components, reflecting consistent performance. In contrast, MLP has relatively poor recall performance, with values starting at 49.23% at 5 components and increasing to 52.40% at 30 components. LR and NB demonstrate moderate recall improvements, with LR increasing from 54.17% to 58.30% and NB increasing from 49.76% to 52.01%. SVM maintains low and stable recall values, ranging from 53.49% at 5 components to 53.49% at 30 components, showing minimal improvement as the number of components increases. DT shows a gradual increase in recall from 58.58% to 56.07%, reflecting its generally strong performance compared to most models. In general, the recall values are relatively low in all models compared to other feature selection methods, indicating that PCA does not significantly improve the ability of these models to identify true positives. Random Forest is the most effective in terms of recall, whereas other models such as SVM and MLP show relatively poor performance.

Figure 3d illustrates the F-1 scores for various ML models as the number of principal components increases from 5 to 30. The F-1 score, which balances precision and recall, highlights the trade-offs between these two metrics. RF consistently achieves the highest F-1 scores in all principal components, starting at 61.34% with 5 components and increasing to 64.59% with 30 components. This model demonstrates high precision and recall, resulting in the most balanced performance among the models tested. In contrast, KNN shows a stable F-1 score, remaining between 58.67% and 58.74%, indicating consistent performance but at a lower level compared to Random Forest. DT exhibits moderate performance in terms of the F-1 score, with values starting at 58.57% at 5 components and reaching 56.07% at 30 components, showing a slight decrease as the number of principal

components increases. LR shows a steady improvement from 54.03% to 58.30%, reflecting a positive trend in its ability to balance precision and recall. MLP shows relatively poor F-1 scores, with values starting at 44.65% and increasing to 45.33%, indicating that this model struggles to effectively balance precision and recall. NB also shows low F-1 scores, starting at 37.21% and increasing to 45.04%, with a notable but still limited improvement in performance. Finally, SVM maintains low and relatively stable F-1 scores throughout the range of principal components, starting at 51.74% at 5 components and remaining nearly unchanged up to 30 components. This reflects a consistent, yet underwhelming, performance in achieving a balance between precision and recall. In summary, the F-1 score analysis reveals that while RF leads in balanced performance, other models such as KNN, LR, and DT show varying levels of effectiveness, and MLP and NB demonstrate relatively poor performance. The use of PCA for feature selection does not lead to significant improvements in F-1 scores in most models.

Figure 3e displays the AUC-ROC scores for various ML models as the number of principal components increases from 5 to 30. The AUC-ROC score measures the performance of a model in distinguishing between classes, with higher values indicating better performance. RF demonstrates the highest AUC-ROC scores among all models, starting at 68.10% with 5 principal components and increasing to 70.18% with 30 components. This indicates that RF consistently performs well in terms of class separation as more principal components are considered. KNN also performs well in AUC-ROC, showing steady and high performance from 62.60% at 5 components to 62.60% at 30 components. This indicates that KNN maintains strong performance throughout the feature selection process. LR and NB show moderate AUC-ROC scores. LR starts at 55.95% and increases to 61.55%, reflecting a trend of improved performance with more principal components. NB starts at 55.04% and increases to 58.38%, showing a more modest improvement in performance. MLP shows relatively high AUC-ROC values, starting at 48.11% and increasing to 58.47% as the number of principal components increases. This shows a gradual improvement in performance, though it remains lower compared to the top-performing models. DT exhibits the lowest AUC-ROC scores among the models, with values ranging from 54.45% to 56.04%. This indicates that DT struggles more than other models in distinguishing between classes effectively. SVM maintains relatively stable but low AUC-ROC performance, starting at 55.73% and showing minimal variation up to 30 principal components. This indicates that SVM's ability to distinguish between classes is limited compared to other models. In summary, RF leads in AUC-ROC performance, showing the best results among all models as the number of principal components increases. KNN also performs well, while other models like DT and SVM show relatively poor AUC-ROC scores. Most models exhibit some improvement with more principal components, but overall performance is modest compared to other feature selection methods.

Principal Component Analysis (PCA) consistently yielded lower performance compared to other feature selection methods in all classifiers. This underperformance can theoretically be explained by the nature of PCA itself. As an unsupervised dimensionality reduction technique, PCA transforms the original feature space into orthogonal components based on variance, without considering class labels. Although this helps to capture

Table 6: Selected Features by the Genetic Algorithm (GA)

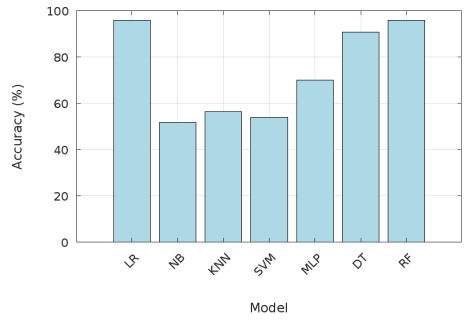
No.	Selected Feature	Category
1	BinaryNumTarget	Meta-data
2	statuses_count	Meta-data
3	following	Meta-data
4	BotScoreBinary	Meta-data
5	normalize_influence	Meta-data
6	mentions	Meta-data
7	replies	Meta-data
8	favourites	Meta-data
9	ORG_percentage	Text
10	DATE_percentage	Text
11	FAC_percentage	Text
12	LAW_percentage	Text
13	PRODUCT_percentage	Text
14	LOC_percentage	Text
15	LANGUAGE_percentage	Text
16	Word count	Text
17	Min word length	Text
18	present_verbs	Lexical
19	past_verbs	Lexical
20	adverbs	Lexical
21	questions	Lexical
22	ampersand	Lexical
23	long_word_freq	Lexical

the overall structure of the data, it does not necessarily preserve the most relevant features for distinguishing between real and fake content. In contrast, methods such as RFE and SelectKBest directly evaluate the relevance of each feature with respect to the classification target, making them more effective in contexts where feature-label relationships are critical.

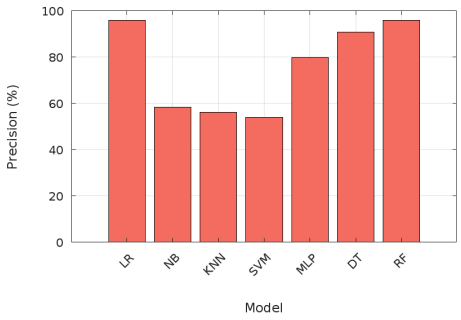
4.3.4. Genetic Algorithm Results

This section presents the results obtained by various ML models, following a feature selection step performed using the GA. The GA algorithm optimally selected 23 features, as shown in Table 6. This set of features was used to train and evaluate multiple ML models, and their performance was assessed based on metrics including accuracy, precision, recall, F1 score, and AUC-ROC, as shown in figure 4. To minimize the risk of overfitting during the selection of features based on genetic algorithms, a 5-fold cross-validation strategy was integrated into the GA fitness evaluation. Following the feature selection process, all classifiers were re-trained and evaluated on a separate unseen test set to ensure robust and unbiased performance assessment.

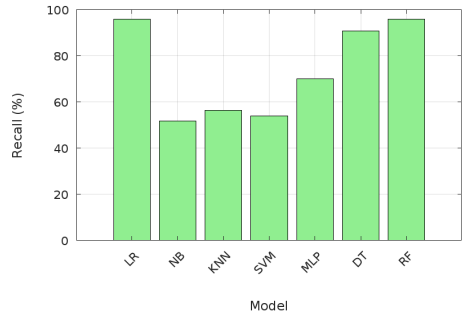
Figure 4a illustrates the accuracy scores of various ML models following the feature selection step performed by the GA algorithm. The DT and RF models exhibit the highest accuracy among all models, achieving nearly 91% and 96%, respectively. LR also shows a high accuracy of approximately 96%, indicating strong performance on the test set. In contrast, the KNN and SVM models demonstrate significantly lower accuracy, with values of around 56% and 54%, respectively.



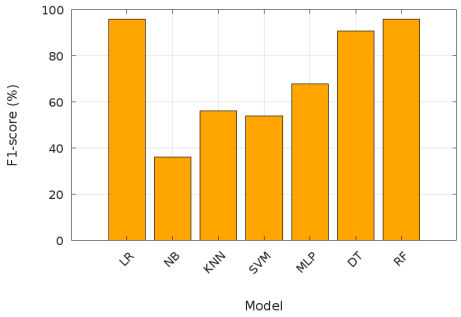
a Accuracy



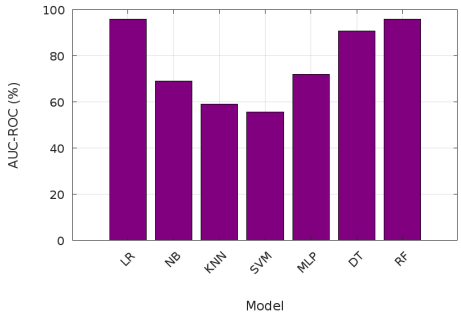
b Precision



c Recall



d F1 Score



e AUC-ROC

Figure 4: Performance Metrics of Various Models under GA Feature Selection.

MLP performs moderately with an accuracy of approximately 70%, while NB shows the lowest accuracy at approximately 52%. Overall, the accuracy results highlight that ensemble methods such as DT and RF perform significantly better compared to other models following the feature selection step. Figure 4b presents the precision scores for

different models after applying the GA feature selection algorithm. LR and RF achieve the highest precision scores of approximately 96% and 96%, respectively, indicating that these models are effective at correctly identifying positive instances. The DT model also shows high precision, around 91%, demonstrating its effectiveness in distinguishing positive cases. MLP shows a good precision of about 80%. In contrast, the KNN and SVM models exhibit lower precision scores, around 56% and 54%, respectively. The NB model achieves a moderate precision score of approximately 58%. This figure underscores that LR, RF, and DT are highly precise, while other models like KNN and SVM show relatively poor performance in terms of precision. Figure 4c shows the recall scores for various models after feature selection using the GA algorithm. LR and RF achieve the highest recall scores, reaching around 96% and 96%, respectively. DT follows closely with a recall score of approximately 91%, indicating an effective identification of positive instances. MLP shows a moderate recall score of about 70%, while KNN and SVM exhibit significantly lower recall scores, approximately 56% and 54%, respectively. NB has the lowest recall score of around 52%. Recall analysis reveals that LR and RF are the most effective in detecting positive instances, while other models such as KNN and SVM perform poorly in this regard.

Figure 4d displays the F1 scores for different ML models after the GA feature selection process. The LR and RF models lead with the highest F1 scores, approximately 96% and 96%, respectively. DT also shows a high F1 score of about 91%, demonstrating balanced performance between precision and recall. MLP exhibits a moderate F1 score of around 68%. In contrast, the KNN and SVM models show relatively low F1 scores, about 56% and 54%, respectively. NB achieves the lowest F1 score of approximately 36%. This figure highlights that the ensemble methods, specifically DT and RF, along with LR, demonstrate superior overall performance compared to other models. Figure 4e presents the AUC-ROC scores for different models following the GA feature selection process. RF achieves the highest AUC-ROC score of approximately 95.76%, demonstrating the best performance in distinguishing between positive and negative instances. LR follows closely with a score of around 95.85%, which also indicates strong performance. On the other hand, KNN and SVM exhibit lower AUC-ROC scores of about 58.99% and 55.53%, respectively, reflecting their relatively weaker performance in class discrimination. NB achieves a moderate AUC-ROC score of approximately 69.07%, while DT shows an AUC-ROC score of approximately 90.72%, which is lower compared to LR and RF but still reflects reasonable performance. In general, the results highlight that LR and RF demonstrate the highest effectiveness in classifying instances, whereas KNN and SVM show relatively poor performance.

4.4. Summary of key insights

The evaluation of feature selection methods—RFE, SelectKBest, PCA, and GA—reveals several important trends in how these techniques influence model performance. This section discusses the general insights gained from these methods and highlights the stability of different ML models across various feature selection techniques.

In general, the findings of this study can be summarized as the following key insights.

- **Performance Stability Across Feature Selection Methods:** RF and LR are notable for their stable performance across different feature selection methods. RF, in particular, shows high and consistent performance across RFE, SelectKBest, and GA methods. For example, RF achieves the highest AUC-ROC score of 95.85% under the GA method, demonstrating its robustness in handling different subsets of features. However, while LR maintains high performance under SelectKBest, it is less consistent under RFE and PCA, indicating that its effectiveness is more dependent on the feature selection process.

- **Effectiveness of Dimensionality Reduction Techniques:** The PCA technique generally proves less effective compared to other feature selection methods. PCA's dimensionality reduction often leads to decreased model performance due to the loss of important feature information and interpretability. This is in contrast to RFE and SelectKBest, which focus on selecting the most relevant features and thereby improve model performance. In this study, PCA fails to effectively enhance model performance, reinforcing the idea that PCA is less suitable for tasks that require feature relevance and interpretability.

  While Principal Component Analysis is well regarded for its dimensionality reduction capabilities, it inherently compromises feature interpretability, which can hinder explainability in applications such as fake news detection. In this study, PCA was intentionally included as a benchmarking tool to examine the trade-offs between dimensionality reduction and interpretability. Although PCA contributed to performance improvement in limited cases, it was consistently outperformed by RFE and GA, which not only yielded better classification results, but also preserved transparency at the feature level. These findings reinforce the practical advantages of using interpretable feature selection techniques for fake news detection.

- **Success of Genetic Algorithms :** GA emerges as a highly effective feature selection technique. The ability of GA to explore a wide range of feature combinations allows it to identify optimal subsets that significantly enhance the performance of the model. This effectiveness is demonstrated by RF achieving the highest AUC-ROC score of 95.85% with the GA method. GA's success highlights its potential as a powerful tool for feature selection, providing a robust approach to optimize feature subsets and improve classification results.

In summary, this study reveals that different feature selection methods have different impacts on model performance. RFE and SelectKBest demonstrate effectiveness in improving model outcomes, whereas PCA tends to degrade performance due to reduced feature relevance and interpretability. Among the models evaluated, Random Forest (RF) consistently shows high reliability and strong performance across all feature selection techniques, highlighting its robustness. Logistic regression (LR) also performs competitively, but exhibits greater sensitivity to the selected feature selection method. Genetic Algorithms (GAs) further prove effective in optimizing feature subsets, as evidenced by the

superior results obtained when combined with RF. These findings emphasize the importance of pairing strong classifiers with interpretable and computationally efficient feature selection strategies for fake news detection tasks. Furthermore, the evolving nature of fake news poses a significant challenge to static models, particularly in the face of adversarial content designed to bypass traditional detection mechanisms. Although the current approach relies on offline learning, these observations point to the need to explore incremental and adaptive machine learning methods that can dynamically adjust to new patterns and threats. Such approaches have the potential to improve the long-term effectiveness and resilience of fake news detection systems.

In addition to technical considerations, ethical implications also play a crucial role in the development and deployment of fake news detection systems. A major concern is the possibility of false positives, cases in which true information is incorrectly flagged as fake, which can suppress valid discourse, damage reputations, and undermine public trust. To address this, the study incorporated a rigorous evaluation framework using a comprehensive set of performance metrics, including precision, recall, F1 score, and AUC-ROC, to ensure a balanced understanding of model behavior. However, technical accuracy alone is not sufficient. Responsible deployment should consider transparency, accountability, and, where appropriate, human oversight. Future work could benefit from incorporating fairness-aware techniques and ethical design principles to ensure that detection systems are not only accurate but also aligned with societal values.

5. Conclusions and Future Work

This study provided a comprehensive analysis of various machine learning classifiers and feature selection techniques to enhance the detection of fake news. Multiple models, including RF, Gradient Boosting, and SVM, were evaluated to identify the strengths and limitations of each approach in the context of detecting fake news. The findings indicated that ensemble learning methods, particularly RF and Gradient Boosting, achieved superior performance in terms of accuracy and robustness. This improvement was especially pronounced when these models were paired with effective feature selection techniques, such as RFE and SelectKBest, which reduced dimensionality and enhanced model performance by focusing on the most relevant features.

The impact of different feature selection methods on the effectiveness of various classifiers was thoroughly examined, demonstrating that techniques like RFE and SelectKBest significantly improved performance. In contrast, PCA was found to be less effective, underscoring the importance of carefully selecting feature extraction methods tailored to the specific application.

In conclusion, the integration of robust ML models with optimized feature selection strategies was shown to be critical to achieving high accuracy and reliability in fake news detection. This study provided valuable information on the development of more effective fake news detection systems and contributed to the advancement of methodologies in this field.

While this study demonstrated the effectiveness of various traditional and ensemble

classifiers combined with feature selection techniques, future research could explore additional avenues to further enhance the detection of fake news. One promising direction involves the investigation of incremental and online learning approaches that allow models to continuously adapt to new data. Given the dynamic and evolving nature of misinformation, such adaptive techniques can improve response to emerging trends without the need for complete retraining. Additionally, evaluating model performance across multiple datasets from different domains and platforms could help assess generalizability. Incorporating explainable AI (XAI) techniques can also strengthen the transparency and trustworthiness of detection systems, particularly in high-stakes information environments. These directions aim to build on the current work and extend its applicability to real-world, evolving contexts.

Another potential extension is the inclusion of a comparative analysis of the computational cost associated with different feature selection techniques in various types of classifiers. Although the present study focused primarily on classification performance and given that the work does not target deployment in resource-constrained environments such as IoT or edge devices, computational efficiency was not emphasized. However, such analysis would be highly valuable in scenarios that require trade-offs between accuracy and efficiency, and could inform deployment decisions in constrained environments.

Acknowledgements

Machine learning training and evaluation have been performed using the Phoenix High Performance Computing facility at the American University of the Middle East, Kuwait.

References

[1] Femi Olan, Uchitha Jayawickrama, Emmanuel Ogiemwonyi Arakpogun, Jana Suklan, and Shaofeng Liu. Fake news on social media: the impact on society. *Information Systems Frontiers*, 26(2):443–458, Apr 2024.

[2] H. Allcott and M. Gentzkow. Social media and fake news in the 2016 election. *Journal of Economic Perspectives*, 31(2):211–236, 2017.

[3] A. Gupta and A. Roy. Manipulation of social media during the 2019 indian general elections. *Asian Journal of Communication*, 29(5):537–556, 2019.

[4] M. Cinelli and A. Galeazzi. The covid-19 social media infodemic. *Scientific Reports*, 10:1–10, 2020.

[5] Omar Abdulwahabe Mohamad Rasha Talal Hameed. Federated learning in iot: A survey on distributed decision making. *Babylonian Journal of Internet of Things*, 2023:1–7, Jan. 2023.

[6] Dipti Theng and Kishor K. Bhoyar. Feature selection techniques for machine learning: a survey of more than two decades of research. *Knowledge and Information Systems*, 66(3):1575–1637, Mar 2024.

[7] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.

[8] Girish Chandrashekar and Ferat Sahin. A survey on feature selection methods. *Computers & Electrical Engineering*, 40(1):16–28, 2014.

[9] Michael Greenacre, Patrick J. F. Groenen, Trevor Hastie, Alfonso Iodice D'Enza, Angelos Markos, and Elena Tuzhilina. Principal component analysis. *Nature Reviews Methods Primers*, 2(1):100, Dec 2022.

[10] Rania Saidi, Waad Bouaguel Ncir, and Nadia Essoussi. Feature selection using genetic algorithm for big data. In *The International Conference on Advanced Machine Learning Technologies and Applications (AMLTA2018)*, pages 352–361, 2018.

[11] Saad Munir and M. Asif Naeem. Bil-fand: leveraging ensemble technique for efficient bilingual fake news detection. *International Journal of Machine Learning and Cybernetics*, Mar 2024.

[12] Ehtesham Hashmi, Sule Yildirim Yayilgan, Muhammad Mudassar Yamin, Subhan Ali, and Mohamed Abomhara. Advancing fake news detection: Hybrid deep learning with fasttext and explainable ai. *IEEE Access*, 12:44462–44480, 2024.

[13] Mahabuba Akhter, Syed Md. Minhaz Hossain, Rizma Sijana Nigar, Srabanti Paul, Khaleque Md. Aashiq Kamal, Anik Sen, and Iqbal H. Sarker. Covid-19 fake news detection using deep learning model. *Annals of Data Science*, Jan 2024.

[14] Chander Prabha, Meena Malik, Shalini Kumari, Neha Arya, Parul Parihar, and Jaspreet Singh. Detection of fake news: A comparative analysis using machine learning. *AIP Conference Proceedings*, 3072(1):040014, 03 2024.

[15] Joy Gorai and Dilip Kumar Shaw. Semantic difference-based feature extraction technique for fake news detection. *The Journal of Supercomputing*, Jun 2024.

[16] Muhammed Baki Çakı and Muhammet Sinan Başarslan. Classification of fake news using machine learning and deep learning. *Journal of Artificial Intelligence and Data Science*, 4(1):22–32, 2024.

[17] Manvi Bohra, Indrajeet Kumar, and Kamred Udham Singh. Fake news detection using different machine learning algorithms. In *2024 2nd International Conference on Device Intelligence, Computing and Communication Technologies (DICCT)*, pages 45–50, 2024.

[18] Pooja Malhotra and S. K. Malik. Fake news detection using ensemble techniques. *Multimedia Tools and Applications*, 83(14):42037–42062, Apr 2024.

[19] Kirandeep Kaur and Nirbhay Kashyap. Fake news detection using boosting ensemble method. In *2024 14th International Conference on Cloud Computing, Data Science Engineering (Confluence)*, pages 532–537, 2024.

[20] Mohammed A. Taha, Haider D. A. Jabar, and Widad K. Mohammed. Fake news detection model basing on machine learning algorithms. *Baghdad Science Journal*, 21(1):35–45, January 2024. Published Online First: January 20, 2024. Received 07/03/2023, Revised 14/07/2023, Accepted 16/07/2023.

[21] Gregorius Airlangga. Comparative analysis of machine learning algorithms for detecting fake news: Efficacy and accuracy in the modern information ecosystem. *Journal of Computer Networks, Architecture and High Performance Computing*, 6(1):354–363, Jan. 2024.

[22] Sheik Abdullah A, Parkavi R, P.Je Sai Kailash, and Deepthi Ramesh. Analysis of on-

line fake news using machine learning techniques. In *2024 Second International Conference on Emerging Trends in Information Technology and Engineering (ICETITE)*, pages 1–5, 2024.

[23] Azka Kishwar and Adeel Zafar. Fake news detection on pakistani news using machine learning and deep learning. *Expert Systems with Applications*, 211:118558, 2023.

[24] Minjung Park and Sangmi Chai. Constructing a user-centered fake news detection model by using classification algorithms in machine learning techniques. *IEEE Access*, 11:71517–71527, 2023.

[25] Mutaz A. B. Al-Tarawneh, Omar Al-irr, Khaled S. Al-Maaitah, Hassan Kanj, and Wael Hosny Fouad Aly. Enhancing fake news detection with word embedding: A machine learning and deep learning approach. *Computers*, 13(9), 2024.

[26] Mutaz A. B. Al-Tarawneh, Omar Alirr, and Hassan Kanj. Performance evaluation of machine learning-based cyber attack detection in electric vehicles charging stations. *International Journal of Advanced Computer Science and Applications*, 16(3), 2025.

[27] Sudhansu R Lenka, Sukant Kishoro Bisoy, Rojalina Priyadarshini, and Mangal Sain. Empirical analysis of ensemble learning for imbalanced credit scoring datasets: a systematic review. *Wireless Communications and Mobile Computing*, 2022(1):6584352, 2022.

[28] Adel Mellit and Soteris Kalogirou. Assessment of machine learning and ensemble methods for fault diagnosis of photovoltaic systems. *Renewable Energy*, 184:1074–1090, 2022.

[29] Mutaz A. B. Al-Tarawneh, Khaled S. Al-Maaitah, and Ashraf Alkhresheh. Adaptive ensemble selection for personalized cardiovascular disease prediction using clustering and feature selection. *International Journal of Advanced Computer Science and Applications*, 16(3), 2025.

[30] Richard A. Berk. *Classification and Regression Trees (CART)*, pages 1–65. Springer New York, New York, NY, 2008.

[31] Ludmila I Kuncheva. *Combining pattern classifiers: methods and algorithms*. John Wiley & Sons, 2014.

[32] Bernhard Scholkopf. Support vector machines: a practical consequence of learning theory. *IEEE Intelligent systems*, 13, 1998.

[33] Thi Kha Nguyen and Thi Phuong Trang Pham. Predicting bankruptcy using machine learning algorithms. *Tp chí Khoa hc và Công ngh-i hc à Nng*, pages 6–9, 2018.

[34] Thiago José Lucas, Inaê Soares De Figueiredo, Carlos Alexandre Carvalho Tojeiro, Alex Marino G De Almeida, Rafał Scherer, José Remo F Brega, João Paulo Papa, and Kelton Augusto Pontara Da Costa. A comprehensive survey on ensemble learning-based intrusion detection approaches in computer networks. *IEEE Access*, 2023.

[35] Pierre Geurts, Damien Ernst, and Louis Wehenkel. Extremely randomized trees. *Machine learning*, 63:3–42, 2006.

[36] Antonio Criminisi et al. Regression forests for efficient anatomy detection and localization in ct studies, sep. 20, 2010, medical computer visions. recognition techniques and applications in medical imaging.

[37] Earum Mushtaq, Aneela Zameer, and Asifullah Khan. A two-stage stacked ensemble

intrusion detection system using five base classifiers and mlp with optimal feature selection. *Microprocessors and Microsystems*, 94:104660, 2022.

[38] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.

[39] Hamoud Aljamaan and Amal Alazba. Software defect prediction using tree-based ensembles. In *Proceedings of the 16th ACM international conference on predictive models and data analytics in software engineering*, pages 1–10, 2020.

[40] Mimusa Azim Mim, Nazia Majadi, and Peal Mazumder. A soft voting ensemble learning approach for credit card fraud detection. *Heliyon*, 10(3), 2024.

[41] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *2008 Eighth IEEE International Conference on Data Mining*, pages 413–422, 2008.

[42] Antonella Mensi and Manuele Bicego. Enhanced anomaly scores for isolation forests. *Pattern Recognition*, 120:108115, 2021.